



Využití micro:bitů ve výuce matematiky

Diplomová práce

Studijní program:

N1101 Matematika

Studijní obory:

Učitelství matematiky pro střední školy

Učitelství informatiky pro střední školy

Autor práce:

Bc. Kristýna Vacková

Vedoucí práce:

Mgr. Daniela Bímová, Ph.D.

Katedra matematiky a didaktiky matematiky





Zadání diplomové práce

Využití micro:bitů ve výuce matematiky

Jméno a příjmení: **Bc. Kristýna Vacková**
Osobní číslo: P20000932
Studijní program: N1101 Matematika
Studijní obory: Učitelství matematiky pro střední školy
Učitelství informatiky pro střední školy
Zadávací katedra: Katedra matematiky a didaktiky matematiky
Akademický rok: 2020/2021

Zásady pro vypracování:

Diplomová práce je zaměřena na jednu z možností využití mezipředmětových vztahů mezi matematikou a informatikou. Tato možnost spočívá v řešení vhodných matematických úloh s užitím programování tzv. micro:bitů. Cíli teoretické části práce jsou představení micro:bitů, dále pak stručné pojednání o programovacích jazycích či internetových aplikacích, pomocí nichž je možné micro:bity programovat. Dalšími cíli teoretické části práce je uvedení základních informací o těch oblastech matematiky, které mohou být i alespoň částečně procvičovány s využitím micro:bitů. Praktická část práce je věnována sestavení souboru vzorově řešených příkladů, u nichž jsou nejen metodicky popsány konkrétní postupy programování micro:bitů, ale i základní principy potřebné k nalezení správného řešení jednotlivých matematických úloh. Kromě vzorově řešených příkladů je možné navrhnout pracovní listy se zadáními matematických úloh, které lze předložit buď žákům 2. stupně základní školy, anebo i žákům střední školy k samostatnému vyřešení s užitím programování micro:bitů. Úrovně náročností navrhovaných úloh se mohou lišit např. procvičoványmi oblastmi matematiky. Dalším cílem práce je praktické ověření řešení sady sestavených matematických úloh v praktické části práce pomocí programování micro:bitů žáky 2. stupně základní školy či studenty střední školy. Získání zpětné vazby o vytváření řešení matematických úloh pomocí programování s micro:bity od zapojených žáků či studentů do praktického ověřování ve formě dotazníkového šetření.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

tištěná/elektronická
Čeština



Seznam odborné literatury:

ODVÁRKO, O. –KADLEČEK, J.: Přehled matematiky – pro základní školy a víceletá gymnázia. PROMETHEUS, Praha 2004. 270 s. ISBN: 80-7196-276-7
PALKOVÁ, M. –ZEMEK, V.: Průvodce matematikou 1 aneb co byste měli znát z numerické matematiky ze základní školy. Didaktis, Brno 2009. 200 str. ISBN 978-80-7358-274-6
PALKOVÁ, M. a kol.: Průvodce matematikou 2 aneb co byste měli znát z geometrie ze základní školy. Didaktis, Brno 2007. 135 str. ISBN 978-80-7358-275-3
PECH, J. –NOVÁK, M.: Robotika pro střední školy: Programujeme micro:bit pomocí Pythonu. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, České Budějovice 2020. 40 s. ISBN 978-80-7394-787-3
VORDERMAN, C.: Help Your Kids with Maths. Dorling Kindersley Limited, London 2014. 264 p. ISBN 978-1-4093-5571-7
sady učebnic a pracovních sešitů z Matematiky pro 2. stupeň ZŠ a pro SŠ
<https://makecode.microbit.org/>
<https://microbit.org/>

Vedoucí práce:

Mgr. Daniela Bímová, Ph.D.
Katedra matematiky a didaktiky matematiky

Datum zadání práce:

8. dubna 2021

Předpokládaný termín odevzdání:

1. května 2022

L.S.

prof. RNDr. Jan Pícek, CSc.
děkan

doc. RNDr. Jana Příhonská, Ph.D.
vedoucí katedry

V Liberci dne 12. května 2021

Prohlášení

Prohlašuji, že svou diplomovou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Jsem si vědoma toho, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má diplomová práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

14. července 2022

Bc. Kristýna Vacková

Poděkování

Velké poděkování patří vedoucí mé diplomové práce Mgr. Daniele Bímové, Ph.D. za odborné a podnětné vedení, věcné připomínky, cenné rady, podporu a velikou trpělivost. Děkuji jí také za mimořádně vstřícné jednání a za čas, který mi věnovala při tvorbě této diplomové práce.

Mé poděkování patří i paní Ing. Jindře Drábkové, Ph.D. za odborné připomínky a komentáře v oblasti programování.

Také děkuji panu Ing. Tomáši Kazdovi za vřelý a ochotný přístup během testování pracovních listů na střední škole a za výbornou zpětnou vazbu k pracovním listům.

Anotace

Diplomová práce je zaměřena na využití mezipředmětových vztahů mezi matematikou a informatikou v podobě řešení vybraných úloh z geometrie s užitím programování tzv. micro:bitu. Představen je mikrorobot s názvem micro:bit, stručně je pojednáno o programovacích jazycích, pomocí nichž je možné micro:bit programovat, uvedeny jsou základní informace o tématech z geometrie, které mohou být i alespoň částečně procvičovány s využitím micro:bitu. Předloženy jsou pracovní listy se zadáními úloh z geometrie určené k samostatnému vypracování s užitím programování micro:bitu jak pro žáky 2. stupně základní školy, tak i pro žáky střední školy. Popsáno je praktické ověření řešení úloh z pracovních listů a jsou shrnuty výsledky dotazníkového šetření zjišťujícího zpětnou vazbu od žáků k vytvořeným pracovním listům.

Klíčová slova

Micro:bit; programování; geometrie; mezipředmětové vztahy

Annotation

This diploma thesis is focused on the use of interdisciplinary relations between Maths and IT in the form of solving the chosen geometric problems by programming the so-called micro:bit. The micro-robot named micro:bit is introduced and programming languages that can be used to program the micro:bit are briefly discussed. In the thesis, basic information on the used topics from geometry, which can be at least partly practiced by use of the micro:bit, can be found. There are also presented worksheets containing geometric tasks which are intended for independent development using micro:bit for both pupils of elementary school and pupils of secondary school. The practical verification of solutions to the tasks from the worksheets is described and the results of a questionnaire survey determining feedback from pupils on the created worksheets are summarized.

Keywords

Micro:bit; programming; geometry; interdisciplinary relations

Obsah

Seznam tabulek	11
Seznam grafů	11
Seznam obrázků	11
Seznam symbolů a zkratk	15
Úvod	16
1 Geometrická témata	18
1.1 Geometrické zobrazení v rovině	18
1.2 Shodná zobrazení v rovině	19
1.2.1 Osová souměrnost	19
1.2.2 Středová souměrnost	21
1.2.3 Posunutí	23
1.2.4 Otočení	25
1.3 Analytická geometrie v rovině	27
1.3.1 Souřadnicový systém	27
1.3.2 Body	30
1.3.3 Úsečky a vektory	31
1.3.4 Přímký	35
2 Programování	37
2.1 Programovací jazyk	37
2.2 Blokově orientované programování	37
2.3 JavaScript	37
2.4 Python	38
2.5 Témata v programování	38
2.5.1 Základní matematické operace a vypisování	38
2.5.2 Proměnná a základní datové typy	39

2.5.3	Podmínky a logika.....	43
2.5.4	Cykly	46
2.5.5	Funkce	48
3	Micro:bit.....	51
3.1	Základní charakteristika	51
3.2	Vývojové prostředí	52
3.3	Technické požadavky pro práci s micro:bity a pracovními listy.....	56
4	Mezipředmětové vztahy.....	57
4.1	Využití micro:bitu v matematice.....	57
4.2	Využití micro:bitu v dalších školních předmětech	58
5	Pracovní listy a dotazník.....	59
5.1	Tvorba pracovních listů	59
5.1.1	Pracovní listy pro žáky základní školy.....	60
5.1.2	Pracovní listy pro žáky střední školy	61
5.2	Vzorová řešení úloh z pracovních listů.....	62
5.2.1	Vzorová řešení úloh z pracovních listů určených pro žáky ZŠ.....	62
5.2.2	Vzorová řešení úloh z pracovních listů určených pro žáky SŠ.....	69
5.3	Testování pracovních listů.....	85
5.3.1	Testování pracovních listů uskutečněné na ZŠ.....	85
5.3.2	Testování pracovních listů uskutečněné na SŠ.....	86
5.4	Vybraná žákovská řešení úloh z pracovních listů.....	88
5.4.1	Vybraná žákovská řešení úloh z pracovních listů určených pro žáky ZŠ. 90	
5.4.2	Vybraná žákovská řešení úloh z pracovních listů určených pro žáky SŠ. 97	
5.5	Tvorba dotazníku	99
5.5.1	Základní principy tvorby dotazníků.....	100
5.5.2	Pilotní dotazníkové šetření	101
5.5.3	Zpětná vazba.....	101

5.6	Změny a úpravy pracovních listů	106
5.7	Metodická příručka	107
5.7.1	Sledované cíle.....	108
5.7.2	Pomůcky a nástroje	110
5.7.3	Motivace.....	110
5.7.4	Pracovní postupy.....	110
5.7.5	Mezipředmětové vztahy.....	111
5.7.6	Organizační doporučení.....	111
6	Propedeutika	112
6.1	GeoGebra	112
6.2	Geoboard	112
6.3	Geo:bit.....	113
	Závěr	114
	Zdroje	116
	Přílohy.....	118
	Pracovní listy	118
	Dotazník	118

Seznam tabulek

Tabulka 1: Ukázka zápisu dvou operátorů – inkrementace a dekrementace	48
Tabulka 2: Základní údaje	101
Tabulka 3: Počty vypracovaných pracovních listů pro jednotlivá témata z geometrie	102

Seznam grafů

Graf 1: Počty vyplněných pracovních listů v jednotlivých testovaných třídách	102
Graf 2: Procentuální zastoupení (ne)porozumění zadání úloh žáky	103
Graf 3: Procentuální zastoupení (ne)porozumění zadání úloh žáky základní školy ..	104
Graf 4: Procentuální zastoupení (ne)porozumění zadání úloh žáky střední školy	104
Graf 5: Procentuální zastoupení vyjádření žáků k obtížnostem zadaných úloh	105
Graf 6: Procentuální zastoupení vyjádření žáků ZŠ k obtížnostem zadaných úloh	105
Graf 7: Procentuální zastoupení vyjádření žáků SŠ k obtížnostem zadaných úloh	106

Seznam obrázků

Obrázek 1: Ukázka zobrazení bodů X, A v daném rovinném zobrazení	18
Obrázek 2: Ukázka zobrazení dvou různých bodů A, B ve shodném zobrazení	19
Obrázek 3: Ukázka zobrazení bodů v osové souměrnosti s osou o	20
Obrázek 4: Ukázka zobrazení trojúhelníku ABC v osové souměrnosti s osou o	21
Obrázek 5: Ukázka zobrazení bodů ve středové souměrnosti se středem S	22
Obrázek 6: Ukázka zobrazení přímky AB neprocházející středem souměrnosti S v dané středové souměrnosti	22
Obrázek 7: Ukázka zobrazení bodu X ve středové souměrnosti složené ze dvou osových souměrností s osami o_1, o_2	23
Obrázek 8: Ukázka zobrazení bodu X v posunutí daném orientovanou úsečkou PP'	24
Obrázek 9: Ukázka zobrazení bodu A v posunutí vzniklém složením dvou osových souměrností s osami o_1, o_2	24
Obrázek 10: Ukázka zobrazení bodů v otočení se středem S a orientovaným úhlem δ	25

Obrázek 11: Ukázka zobrazení přímky $p \equiv YX$ v otočení se středem otočení S a s orientovaným úhlem α	26
Obrázek 12: Ukázka zobrazení bodu X v otočení složeném ze dvou osových souměrností s osami o_1, o_2	27
Obrázek 13: Zobrazení reálných číselných os x a y kartézské soustavy souřadnic	28
Obrázek 14: Ukázka rozdělení dvourozměrné roviny na čtyři kvadranty	29
Obrázek 15: Ukázka zobrazení souřadnicového systému v micro:bitu	30
Obrázek 16: Úsečka AB s jejím vnitřním bodem X	31
Obrázek 17: Zobrazení úsečky AB v kartézské soustavě souřadnic	31
Obrázek 18: Grafické sčítání vektorů	33
Obrázek 19: Znázornění opačného vektoru k vektoru u	34
Obrázek 20: Grafický rozdíl vektorů u a v	34
Obrázek 21: Ukázka násobení vektoru reálným číslem	35
Obrázek 22: Zobrazení přímky AB a dvou zvolených reprezentantů jejího směrového vektoru u	36
Obrázek 23: Ukázka blokově orientovaného jazyka micro:bit – makecode	37
Obrázek 24: Blokově orientované programování – základní matematická operace (sčítání)	39
Obrázek 25: Blokově orientované programování – záložka „Proměnné“	41
Obrázek 26: Blokově orientované programování – textové a číselné datové typy	41
Obrázek 27: Seznamy v blokově orientovaném programování	42
Obrázek 28: Ukázka podmíněného příkazu v blokově orientovaném programování	43
Obrázek 29: Relační operátory v blokově orientovaném programování	44
Obrázek 30: Logické operátory v blokově orientovaném programování	44
<i>Obrázek 31: Ukázky bloků s cykly v blokově orientovaném programování</i>	<i>47</i>
Obrázek 32: Ukázka definování funkce v blokově orientovaném programování	49
Obrázek 33: Volání funkce v blokově orientovaném programování	49
Obrázek 34: Ukončující blok návratové hodnoty v blokově orientovaném programování	49
Obrázek 35: Micro:bit	51
Obrázek 36: Micro:bit s redukcí	52
Obrázek 37: Ukázka části kódu v blokově orientovaném programování	53
Obrázek 38: Ukázka části kódu zapsaného v JavaScript	53

Obrázek 39: Ukázka části kódu zapsaného v Pythonu	53
Obrázek 40: Vývojové prostředí www.makecode.microbit.org	54
Obrázek 41: Propojení micro:bitu s počítačem I.....	55
Obrázek 42: Propojení micro:bitu s počítačem II	55
Obrázek 43: Vytvořený kód pro simulaci hodu kostkou pomocí micro:bitu.....	58
Obrázek 44: Pracovní list – Úvodní hodina.....	59
Obrázek 45: Ukázka struktury pracovního listu určeného pro žáky základní školy	60
Obrázek 46: Ukázka pracovního listu určeného pro žáky střední školy	61
Obrázek 47: Ukázka řešení 1. úlohy v pracovním listě Osová souměrnost.....	62
Obrázek 48: Ukázka řešení 2. úlohy v pracovním listě Osová souměrnost.....	63
Obrázek 49: Ukázka řešení 3. úlohy v pracovním listě Osová souměrnost.....	63
Obrázek 50: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu v názvem Osová souměrnost	64
Obrázek 51: Ukázka řešení 1. úlohy v pracovním listě Středová souměrnost	64
Obrázek 52: Ukázka řešení 2. úlohy v pracovním listě Středová souměrnost	65
Obrázek 53: Ukázka řešení 3. úlohy v pracovním listě Středová souměrnost	65
Obrázek 54: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem Středová souměrnost	66
Obrázek 55: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem Posunutí	67
Obrázek 56: Ukázka řešení 1. úlohy v pracovním listě Otočení.....	68
Obrázek 57: Ukázka řešení 2. úlohy v pracovním listě Otočení.....	68
Obrázek 58: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem Otočení.....	69
Obrázek 59: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Osová souměrnost.....	70
Obrázek 60: Ukázka vyznačení zadání a řešení v micro:bitu - 2. úloha v pracovním listě Osová souměrnost.....	71
Obrázek 61: Ukázka vyznačení zadání a řešení v micro:bitu - 3. úloha v pracovním listě Osová souměrnost.....	73
Obrázek 62: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Středová souměrnost.....	75

Obrázek 63: Ukázka vyznačení zadání a řešení v micro:bitu - 2. úloha v pracovním listě	
Středová souměrnost.....	77
Obrázek 64: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě	
Otočení.....	79
Obrázek 65: Ukázka vyznačení bodů v micro:bitu	80
Obrázek 66: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě	
Analytická geometrie	82
Obrázek 67: Vybrané žákovské řešení 1. úlohy z pracovního listu Úvodní hodina	88
Obrázek 68: Vybrané žákovské řešení 2. úlohy z pracovního listu Úvodní hodina	88
Obrázek 69: Vybrané žákovské řešení 3. úlohy z pracovního listu Úvodní hodina	89
Obrázek 70: Vybrané žákovské řešení 4. úlohy z pracovního listu Úvodní hodina	89
Obrázek 71: Vybrané žákovské řešení 1. úlohy z pracovního listu Osová souměrnost	90
Obrázek 72: Vybrané žákovské řešení 2. úlohy z pracovního listu Osová souměrnost	91
Obrázek 73: Vybrané žákovské řešení 3. úlohy z pracovního listu Osová souměrnost	91
Obrázek 74: Vybrané žákovské řešení 1. úlohy z pracovního listu Středová souměrnost....	
.....	92
Obrázek 75: Vybrané žákovské řešení 2. úlohy z pracovního listu Středová souměrnost	
.....	92
Obrázek 76: Vybrané žákovské řešení 3. úlohy z pracovního listu Středová souměrnost	
.....	93
Obrázek 77: Vybrané žákovské řešení 1. úlohy z pracovního listu Posunutí.....	94
Obrázek 78: Vybrané žákovské řešení 2. úlohy z pracovního listu Posunutí.....	94
Obrázek 79: Vybrané žákovské řešení 3. úlohy z pracovního listu Posunutí.....	95
Obrázek 80: Ukázka řešení úlohy číslo 3 z pracovního listu Posunutí.....	96
Obrázek 81: Vybrané žákovské řešení 1. úlohy z pracovního listu Otočení	96
Obrázek 82: Vybrané žákovské řešení 2. úlohy z pracovního listu Otočení	96
Obrázek 83: Vybrané žákovské řešení 3. úlohy z pracovního listu Otočení	97
Obrázek 84: Vybrané žákovské řešení 1. úlohy z pracovního listu Osová souměrnost	98
Obrázek 85: Vybrané žákovské řešení 1. úlohy z pracovního listu Osová souměrnost	99
Obrázek 86: Micro:bit makecode – projekty	107
Obrázek 87: Ukázka zobrazení čtverce v osově souměrnosti s vyznačenou osou	
souměrnosti na Geoboardu	113
Obrázek 88: Geo:bit.....	113

Seznam symbolů a zkratek

Symbol	Význam	Příklad použití
$\in$	Je prvkem množiny	$A \in p$ – bod A leží na přímce p
$\notin$	Není prvkem množiny	$A \notin q$ – bod A neleží na přímce q
$ \cdot\cdot $	Vzdálenost mezi dvěma body, velikost / délka úsečky	$ AB $ – vzdálenost mezi dvěma různými body A, B / délka úsečky AB
$\equiv$	Totožnost	$A \equiv B$ – bod A je totožný s bodem B / bod A splývá s bodem B
$\neq$	Různost	$A \neq B$ – bod A je různý od bodu B / bod A nesplývá s bodem B
$? [\cdot ; \cdot]$	Souřadnice bodu v kartézské soustavě souřadnic	$A [a_1; a_2]$ – souřadnice a_1, a_2 bodu A v kartézské soustavě souřadnic
$? (\cdot ; \cdot)$	Souřadnice vektoru v kartézské soustavě souřadnic	$\mathbf{u} (u_1; u_2)$ – souřadnice u_1, u_2 vektoru $\mathbf{u}$ v kartézské soustavě souřadnic
$\rightarrow$	Přiřazení / zobrazení	$X \rightarrow X'$ – bodu X je přiřazen bod X' / bod X se zobrazuje na bod X'
$\circ$	Skládání zobrazení	$A \circ B$ – skládání zobrazení A a B
I	Identita	$I: U \rightarrow U$ – objektu U je v daném zobrazení přiřazen tentýž objekt U

Úvod

Tato diplomová práce se věnuje tématu využití micro:bitů ve výuce matematiky. V rámci jejího vypracování byly vytvořeny speciální pracovní listy obsahující úlohy k procvičení vybraných témat z geometrie. Přitom úlohy z geometrie jsou řešeny dvojím způsobem, a to jednak na základě konstrukčních dovedností a geometrických znalostí žáků, ale také pomocí programování micro:bitu.

Práce je uvedena zmíněním základních poznatků z těchto témat geometrie, pro něž jsou v pracovních listech navrženy příklady k procvičení. Jedná se o shodná zobrazení v rovině a o základy analytické geometrie v rovině. Následuje kapitola představující programovací jazyky, pomocí nichž lze micro:bit programovat. V této kapitole se také nachází přehled témat z programování, která lze využít při řešení úloh zařazených v pracovních listech.

Po představení poznatků z geometrie a programování je v práci zařazena kapitola, která seznamuje s micro:bitem a také s vývojovým prostředím <https://makecode.microbit.org/>. Na tuto kapitolu plynule navazuje kapitola věnovaná mezipředmětovým vztahům, protože jedním z cílů této práce je právě propojení matematiky a informatiky.

Předposlední kapitola se zabývá speciálně vytvořenými pracovními listy a dotazníkem. V úvodu této kapitoly jsou popsány tvorba pracovních listů a vzorová řešení úloh, které jsou v nich zařazeny, následují části o tvorbě dotazníku, testování pracovních listů, na něž navazují zpětná vazba a popis provedených změn v pracovních listech na základně zpětné vazby získané z dotazníkového šetření. Poslední část této kapitoly je věnována představení metodické příručky sepsané k pracovním listům, v níž jsou poznamenána doporučení pro práci s pracovními listy. Poslední kapitolou je propedeutika práce s micro:bity.

Tato práce je určena všem, které zajímá programování, micro:bity a propojení matematiky a informatiky. Také věřím, že lze práci využít jako doprovodný učební materiál, protože se v ní vyskytují poznatky z geometrie, ale také poznatky z programování.

Téma využití micro:bitů ve výuce matematiky jsem si zvolila z toho důvodu, protože matematika a informatika jsou si velmi blízké a sepsání diplomové práce na toto téma bylo jedinečnou příležitostí, jak tyto dva předměty propojit a vytvořit mezipředmětový materiál, který může rozvíjet hned několik oblastí.

Tato práce mi také poskytla možnost vytvoření pracovních listů, které mohu využívat dále během své budoucí učitelské praxe. Jejich zařazováním ve výuce lze s výhodou

propojovat vyučovaná témata z matematiky a informatiky, tím ukázat žákům úzkou vazbu těchto dvou předmětů a případně nejen jich. A to především z toho důvodu, aby na školní předměty nenahlíželi jako na izolované a samostatné oblasti.

1 Geometrická témata

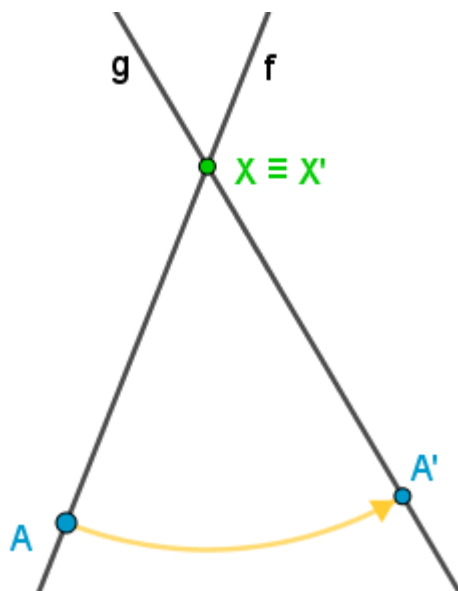
Tato kapitola je věnovaná vybraným tématům z geometrie. Ve speciálně sestavených pracovních listech jsou geometrické úlohy z vybraných témat řešeny nejen konstrukčně pomocí pravítka a kružítka, ale také pomocí programů sestavených pro micro:bity.

1.1 Geometrické zobrazení v rovině

Geometrickým zobrazením v rovině se rozumí předpis, který libovolnému bodu A (vzoru) dané roviny přiřadí jako jeho obraz právě jeden bod A' z té samé roviny.

Pokud se bod X zobrazuje sám na sebe, tj. obraz bodu splývá se svým vzorem, pak takový bod označujeme jako samodružný bod daného zobrazení a zapisujeme $X \equiv X'$. ([1]; 215)

Nechť je dán geometrický útvar U a nechť U' je jeho obraz v daném zobrazení. Pokud je každý bod útvaru U současně i bodem útvaru U' , pak útvar U' splývá s útvarem U . Takový útvar U se v daném zobrazení nazývá samodružný. Je-li každý bod samodružného útvaru U samodružný, pak je útvar U silně samodružný, v opačném případě se nazývá slabě samodružný. Zobrazení, v němž je každý bod samodružný, nazýváme identita a značíme jej I .



Obrázek 1: Ukázka zobrazení bodů X , A v daném rovinném zobrazení

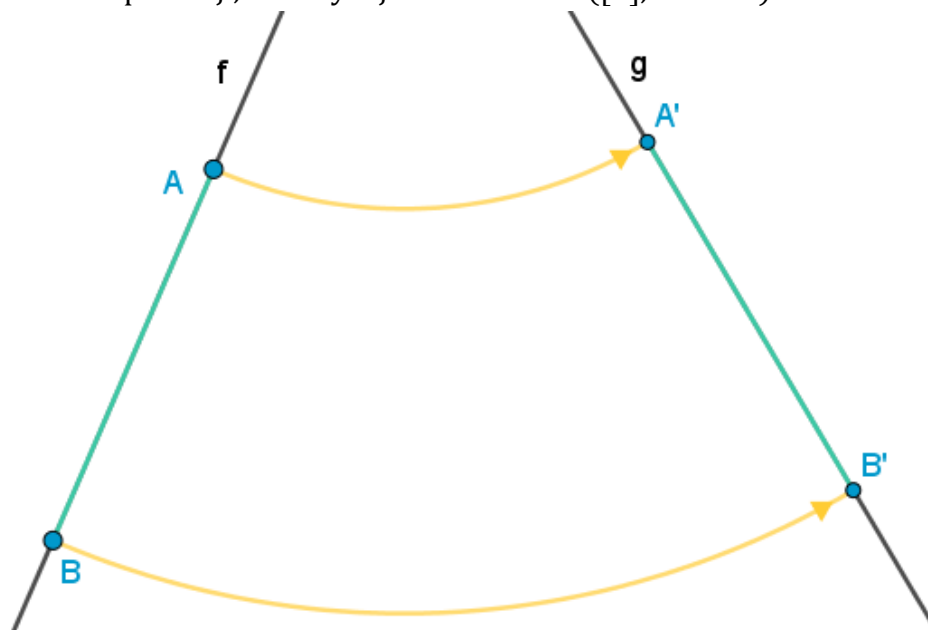
Vztahy, které se v daném zobrazení nemění, nazýváme invarianty, jedná se například o velikosti úseček, úhlů nebo smysl obíhání vrcholů v trojúhelníku.

Ke každému prostému (tj. různým vzorům se přiřazují různé obrazy) zobrazení Z můžeme sestavit tzv. inverzní zobrazení Z^{-1} , které je dáno vztahem

$Z \circ Z^{-1} = Z^{-1} \circ Z = I$, kde symbol „ $\circ$ “ značí skládání zobrazení a I označuje identitu. Tzn. složíme-li zobrazení Z a k němu inverzní zobrazení Z^{-1} , získáme identitu I .

1.2 Shodná zobrazení v rovině

Shodná zobrazení v rovině jsou taková zobrazení, která každým dvěma různým bodům A, B roviny přiřazují dva různé body A', B' roviny tak, že $|AB| = |A'B'|$. Útvary, které si v daném zobrazení odpovídají, se nazývají vzor a obraz. ([1], str. 215)



Obrázek 2: Ukázka zobrazení dvou různých bodů A, B ve shodném zobrazení

Shodné zobrazení zachovává délky úseček a velikosti úhlů. ([10], str. 65)

Shodná zobrazení dělíme na:

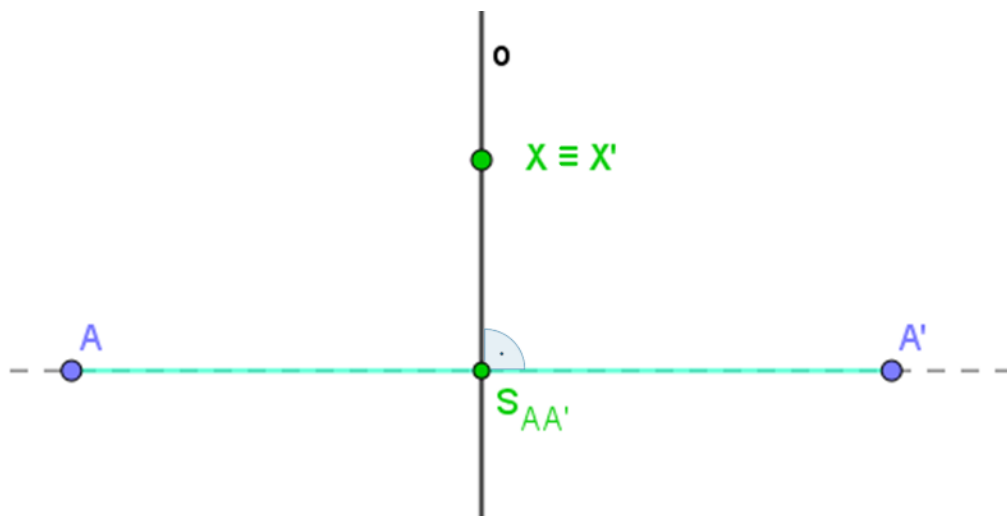
- *přímá* – orientace vnitřních úhlů vzoru a obrazu zůstávají stejné; příklady přímých shodných zobrazení jsou například otočení (rotace), posunutí (translace) a středová souměrnost
- *nepřímá* – orientace vnitřních úhlů vzoru a obrazu jsou opačné; příkladem nepřímého shodného zobrazení je například osová souměrnost

1.2.1 Osová souměrnost

Osová souměrnost je geometrické zobrazení v rovině, které je určené pevně zvolenou přímkou o a ve kterém je ke každému bodu

- $X \in o$ přiřazen bod $X' \equiv X$,
- $A \notin o$ přiřazen bod A' takový, že přímka AA' je kolmá k přímce o a průsečík přímek AA' a o je středem úsečky AA' .

Přímku o nazýváme osa souměrnosti. V osové souměrnosti jsou samodružné všechny body, které leží na ose souměrnosti. ([1]; str. 216)



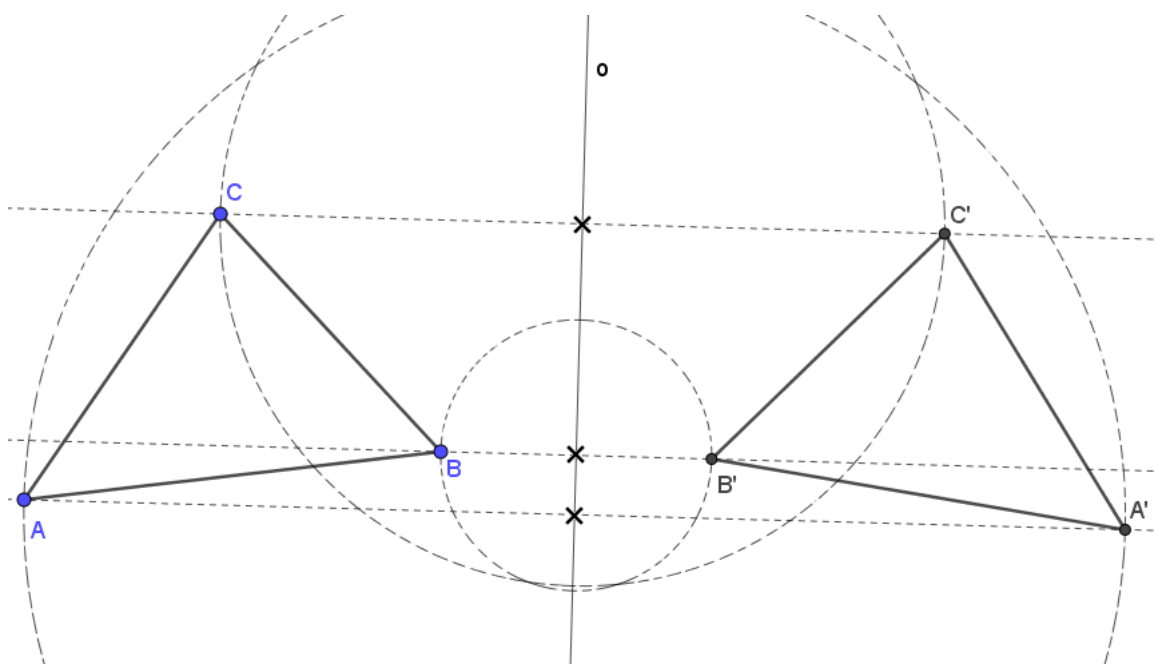
Obrázek 3: Ukázka zobrazení bodů v osové souměrnosti s osou o

Základní principy:

- Osová souměrnost je jednoznačně určena svou osou, nebo dvojicí odpovídajících si bodů (vzorem a obrazem).
- Obrazem bodu A je bod A' . Pokud tento bod leží na ose souměrnosti o , splývá se svým obrazem a je samodružný.
- Samodružné body jsou všechny body ležící na ose souměrnosti.
- Samodružné přímky dělíme v osové souměrnosti na silně samodružné (osa souměrnosti) a slabě samodružné (všechny přímky kolmé k ose souměrnosti).

Základní vlastnosti:

- Obrazem úsečky AB je úsečka $A'B'$, která je shodná s úsečkou AB , přitom osa souměrnosti o je osou úseček AA' a BB' .
- Obrazem přímky AB je přímka $A'B'$. Pokud přímka AB není rovnoběžná s osou souměrnosti o , pak se vzor a obraz protínají na ose souměrnosti o .
- Zobrazení bodu X v osové souměrnosti s osou o symbolicky zapisujeme: $O(o): X \rightarrow X'$.
- Osová souměrnost je příkladem nepřímého shodného zobrazení, tj. mění smysl obíhání vrcholů trojúhelníku. ([8]; str. 87)



Obrázek 4: Ukázka zobrazení trojúhelníku ABC v osové souměrnosti s osou o

Osově souměrné útvary jsou takové útvary, které se v dané osové souměrnosti zobrazují samodružně. Osově souměrnými útvary jsou například čtverec, obdélník, rovnostranný trojúhelník, rovnoramenný trojúhelník, kruh, pravidelný šestiúhelník, ... Přitom každý z uvedených osově souměrných útvarů má různý počet os souměrnosti. Např. čtverec má 4 osy souměrnosti (2 přímky, na nichž leží jeho dvě úhlopříčky, a 2 přímky, na nichž leží střední příčky uvažovaného čtverce); obdélník má 2 osy souměrnosti, a to přímky, které jsou incidentní se středními příčkami obdélníku; rovnostranný trojúhelník má právě 3 osy souměrnosti, které splývají s osami jeho stran; rovnoramenný trojúhelník má právě jednu osu souměrnosti, kterou je osa jeho základny; kruh má nekonečně mnoho os souměrnosti, a to všechny přímky procházející jeho středem atd.

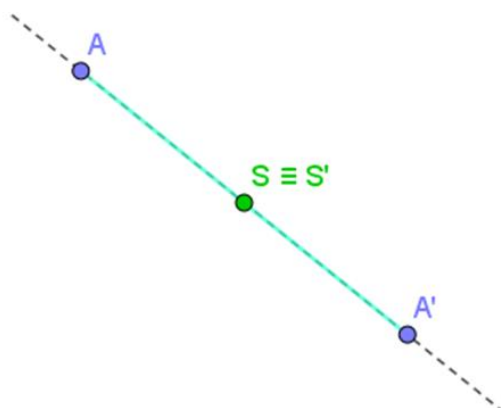
Každé shodné zobrazení v rovině se dá složit z nejvýše tří osových souměrností. Každé přímé shodné zobrazení v rovině lze složit právě ze dvou osových souměrností, nepřímé shodné zobrazení v rovině je pak možné složit ze tří osových souměrností.

1.2.2 Středová souměrnost

Středová souměrnost je geometrické zobrazení v rovině, které je určeno bodem S a ve kterém platí

- obrazem S' bodu S je opět bod S , tj. $S \equiv S'$;
- každému bodu $A \neq S$ je přiřazen bod A' takový, že bod S je středem úsečky AA' .

Bod S se nazývá střed souměrnosti. ([1]; str. 219)



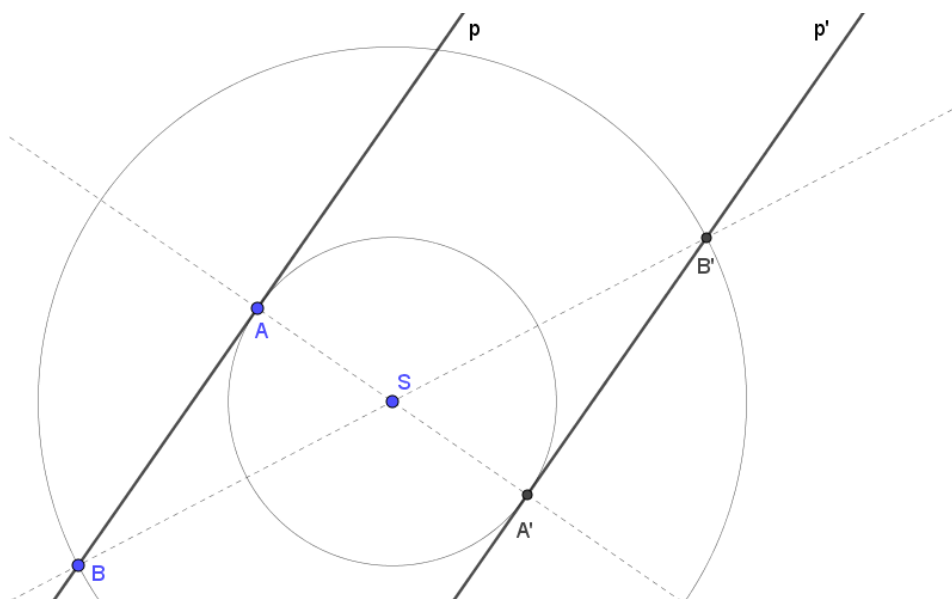
Obrázek 5: Ukázka zobrazení bodů ve středové souměrnosti se středem S

Základní principy:

- Středová souměrnost je jednoznačně určena svým středem, nebo dvojicí odpovídajících si bodů (vzorem a obrazem).
- Ve středové souměrnosti je jediný samodružný bod a tím je střed souměrnosti S .
- Ve středové souměrnosti není ani jedna silně samodružná přímka, na druhou stranu ve středové souměrnosti existuje nekonečně mnoho slabě samodružných přímek. Jsou jimi všechny přímky procházející středem souměrnosti S .

Základní vlastnosti:

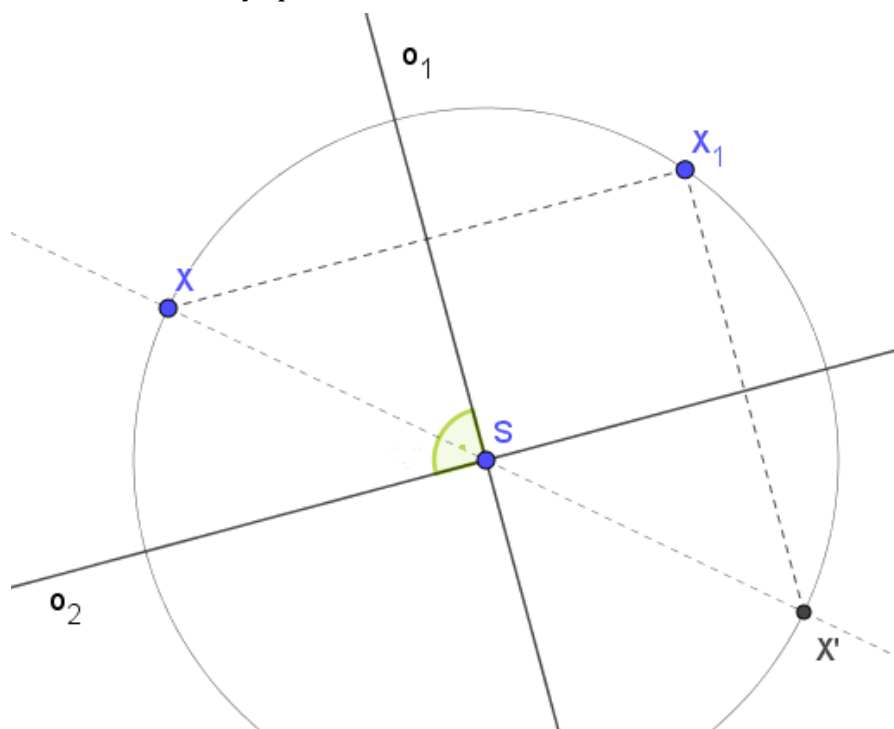
- Obrazem přímky AB je přímka $A'B'$. Pokud tato přímka neprochází středem souměrnosti S , je přímka $A'B'$ rovnoběžná s přímkou AB . ([8]; str. 81)



Obrázek 6: Ukázka zobrazení přímky AB neprocházející středem souměrnosti S v dané středové souměrnosti

- Středová souměrnost je příkladem přímého shodného zobrazení.
- Zobrazení bodu X ve středové souměrnosti se středem souměrnosti S symbolicky zapisujeme: $S(S): X \rightarrow X'$.

Středovou souměrnost je možné složit ze dvou osových souměrností. Osy těchto osových souměrností musí být přitom k sobě kolmé.



Obrázek 7: Ukázka zobrazení bodu X ve středové souměrnosti složené ze dvou osových souměrností s osami o_1, o_2

Středově souměrné útvary jsou takové útvary, které se v dané středové souměrnosti zobrazují samodruzně.

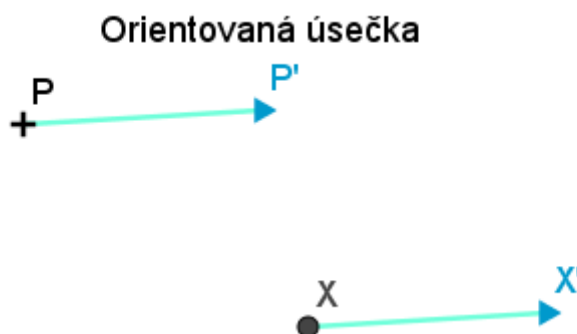
Mezi středově souměrné rovinné útvary patří například kruh, čtverec, obdélník, nebo pravidelný šestiúhelník, ... Přitom středy souměrnosti jsou středy jednotlivých středově souměrných rovinných útvarů.

1.2.3 Posunutí

S posunutím souvisí pojem orientovaná úsečka. Orientovaná úsečka je taková úsečka, u které rozlišujeme počáteční a koncový bod. Počáteční bod bývá v grafickém vyjádření orientované úsečky označován svislou čárkou či křížkem a koncový bod pak šipkou. ([1]; str. 222)

Posunutí nebo také translace je geometrické zobrazení v rovině určené orientovanou úsečkou PP' , která každému bodu X roviny přiřazuje bod X' takový, že orientovaná úsečka

PP' a úsečka XX' jsou shodné (tzn. navzájem rovnoběžné, shodně orientované a stejně dlouhé).



Obrázek 8: Ukázka zobrazení bodu X v posunutí daném orientovanou úsečkou PP'

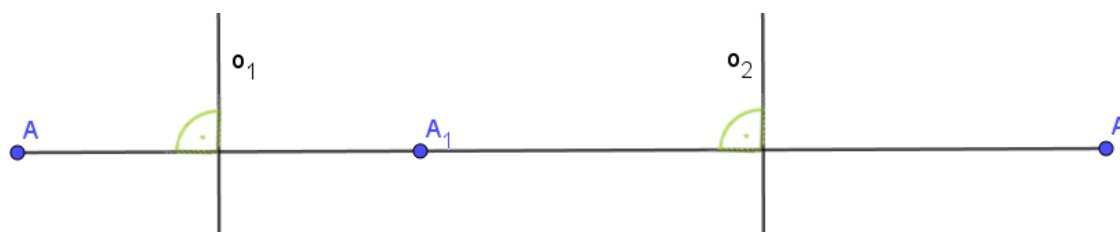
Základní principy:

- Posunutí je jednoznačně určeno orientovanou úsečkou PP' , anebo jednou dvojicí navzájem různých bodů P, P' .
- V posunutí nejsou žádné samodružné body. ([1]; str. 222)
- Slabě samodružné přímky jsou všechny ty, které jsou rovnoběžné se směrem posunutí.

Základní vlastnosti:

- Obrazem úsečky AB je úsečka $A'B'$ shodná a rovnoběžná s úsečkou AB , orientace úseček $AB, A'B'$ je stejná.
- Obrazem přímky AB je přímka $A'B'$ rovnoběžná s přímkou AB .
- Při skládání posunutí nezáleží na pořadí, v jakém dané objekty zobrazujeme. ([8]; str. 92)
- Zobrazení bodu X v posunutí daném orientovanou úsečkou PP' symbolicky zapisujeme: $T(PP'): X \rightarrow X'$.

Každé posunutí lze složit ze dvou osových souměrností, jejichž osy jsou kolmé ke směru posunutí a jsou navzájem rovnoběžné různé. Délka posunutí je pak rovna dvojnásobku vzdálenosti dvou daných os souměrností o_1, o_2 .



Obrázek 9: Ukázka zobrazení bodu A v posunutí vzniklém složením dvou osových souměrností s osami o_1, o_2

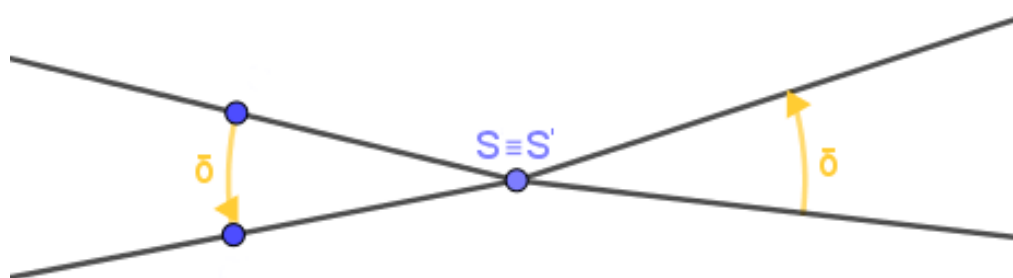
1.2.4 Otočení

S otočením souvisí pojem orientovaný úhel. Orientovaný úhel je takový úhel, ve kterém rozlišujeme počáteční a koncové rameno. V grafickém vyjádření orientovaného úhlu bývá šipkou u oblouku naznačeno, jak je orientovaný, anebo v algebraickém vyjádření orientovaného úhlu jsou před zápisem jeho číselné hodnoty napsána znaménka „+“ či „-“. Znaménko „+“ znamená otočení proti směru hodinových ručiček a znaménko „-“ představuje otočení po směru hodinových ručiček. ([1]; str. 223, 225)

Otočení, nebo také rotace je geometrické zobrazení v rovině určené bodem S a orientovaným úhlem δ , kde

- obrazem S' bodu S je opět bod S , tj. $S \equiv S'$;
- obrazem libovolného bodu $X \neq S$ je takový bod X' , že $|X'S| = |XS|$ a orientované úhly δ a $\sphericalangle XSX'$ jsou shodné a shodně orientované. ([1]; str. 224)

Bod S se nazývá střed otočení. V otočení je jediný samodružný bod a tím je střed otočení S . ([1]; str. 224)



Obrázek 10: Ukázka zobrazení bodů v otočení se středem S a orientovaným úhlem δ

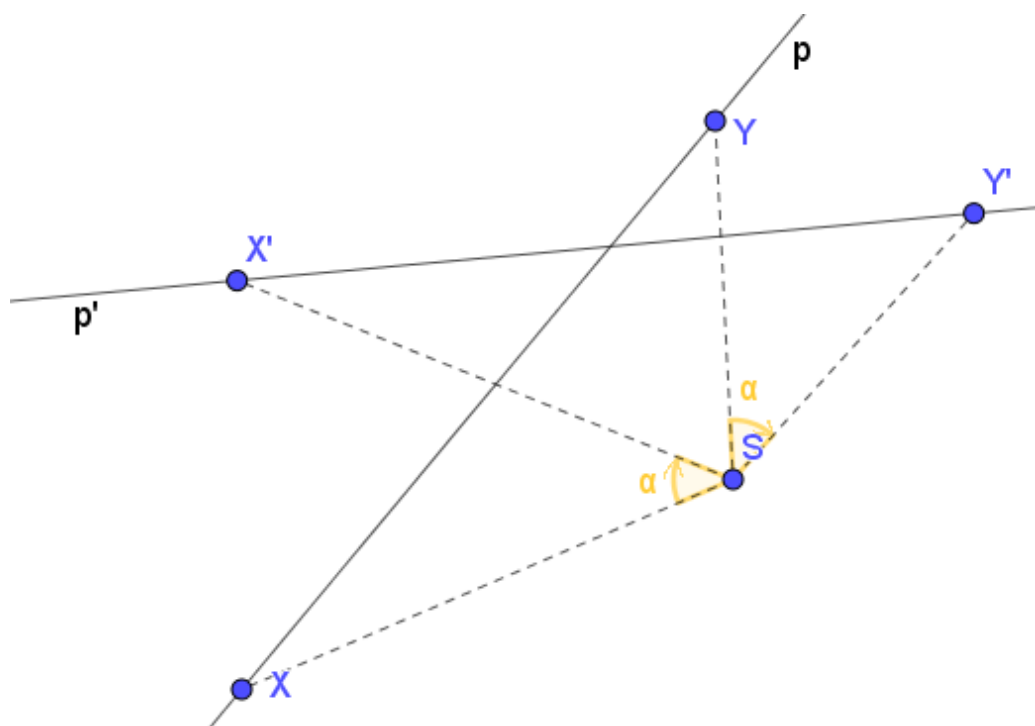
Základní principy:

- Otočení je jednoznačně určeno středem otočení S a současně buď orientovaným úhlem, anebo dvojicí odpovídajících si bodů X, X' .
- Jediným samodružným bodem je střed otočení S .
- Otočení nemá žádné samodružné přímky.

Základní vlastnosti:

- Obrazem úsečky AB je úsečka $A'B'$, která je shodná s úsečkou AB . Přitom platí, že $|SA| = |SA'|$, $|SB| = |SB'|$ a $|\sphericalangle ASA'| = |\sphericalangle BSB'| = \delta$. ([8]; str. 95)

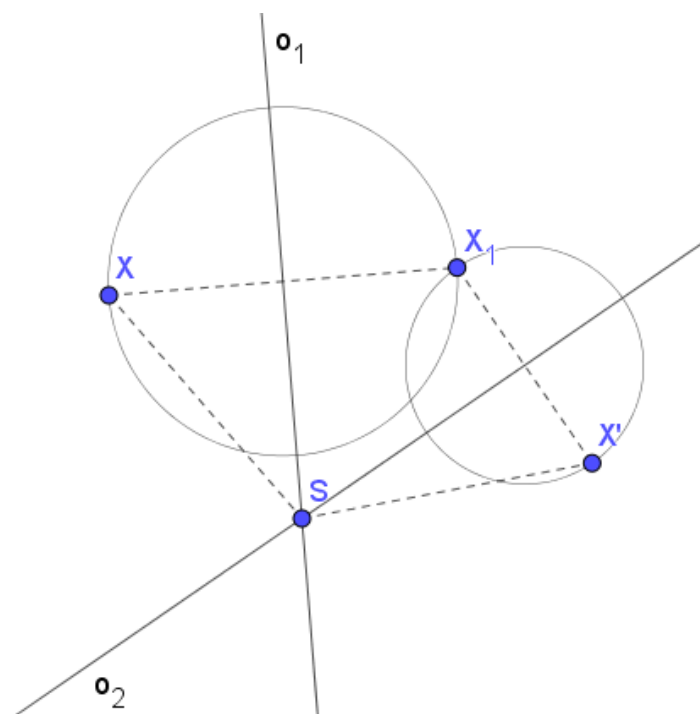
- Otočení se středem S a orientovaným úhlem α (různým od nulového úhlu a každého úhlu, který je celým násobkem plného úhlu) převádí každou přímku na přímku s danou přímkou různoběžnou.



Obrázek 11: Ukázka zobrazení přímky $p \equiv YX$ v otočení se středem otočení S a s orientovaným úhlem α

- Je-li úhel otočení lichým násobkem 180° , dostaneme středovou souměrnost.
- Je-li úhel otočení sudým násobkem 180° , získáme identitu.
- Otočení je příkladem přímého shodného zobrazení.
- Zobrazení bodu X v otočení daném středem otočení S a orientovaným úhlem α symbolicky zapisujeme: $R(S, \alpha): X \rightarrow X'$.

Každé otočení lze složit ze dvou osových souměrností, jejichž osy jsou různoběžné se společným průsečíkem ve středu otočení S . A obráceně, každé otočení lze rozložit na dvě osové souměrnosti s osami navzájem různoběžnými.



Obrázek 12: Ukázka zobrazení bodu X v otočení složeném ze dvou osových souměrností s osami o_1, o_2

1.3 Analytická geometrie v rovině

Základní rozdíl mezi planimetrií, tj. konstrukční geometrií v rovině a analytickou geometrií v rovině spočívá v tom, že v planimetrii nezáleží na přesném umístění geometrických objektů v rovině. Tj. jakákoliv konstrukční úloha bude mít stále stejné řešení nezávisle na umístění geometrických objektů v rovině. Na druhou stranu při řešení úloh v analytické geometrii již závisí na přesném umístění geometrických objektů v rovině. Pro jejich jednotné umístění se v analytické geometrii zavádí kartézský souřadnicový systém.

Programování micro:bitů také využívá zadávání prvků pomocí souřadnic, a proto je možné použít micro:bit i k procvičování analytické geometrie.

1.3.1 Souřadnicový systém

Pro přesný popis polohy objektu ve dvourozměrné rovině nestačí pouze jedna souřadnice, ale potřebujeme k němu souřadnice dvě. Na základě této uvedené potřeby již vzniká systém souřadnic, někdy též označovaný souřadnicový systém. ([2]; str. 8-9)

Existuje mnoho způsobů, jak zaznamenat a vyobrazit umístění nějakého objektu v rovině, některými z nich mohou být např. polární systém souřadnic, zeměpisná soustava

souřadnic apod. V analytické geometrii se využívá především kartézské soustavy souřadnic.

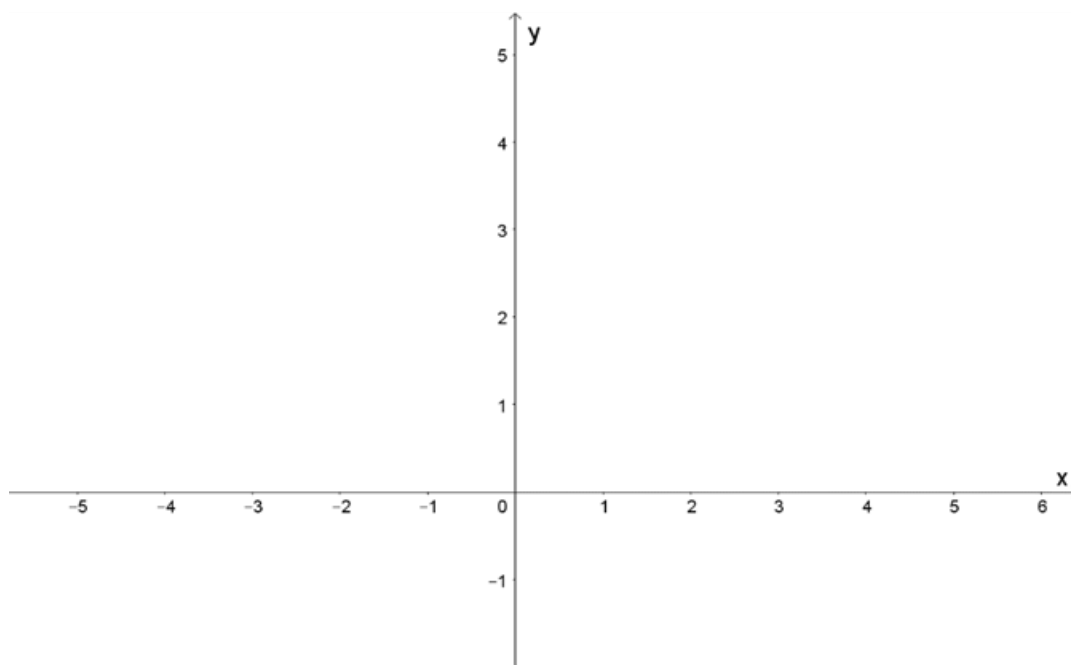
Kartézská soustava souřadnic

Kartézskou soustavou souřadnic rozumíme dvojici reálných číselných os x a y v rovině, kde:

- jsou obě tyto osy k sobě kolmé,
- jejich průsečík O odpovídá na obou osách číslu 0.

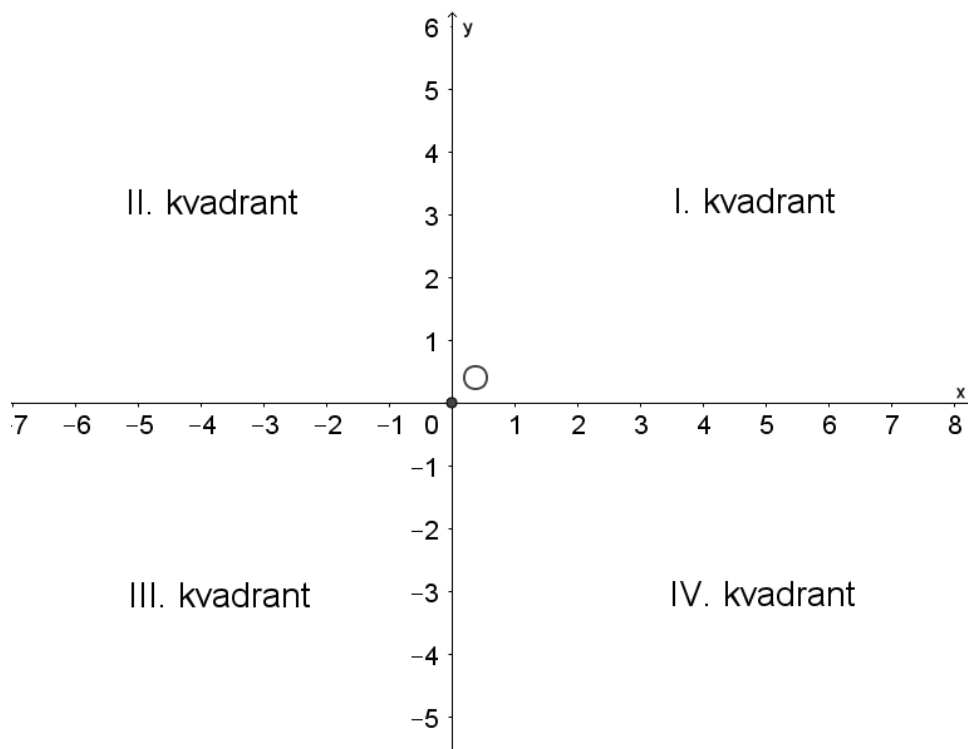
Bod O je počátek kartézské soustavy souřadnic a přímky x, y jsou souřadnicové osy této soustavy. ([4]; str. 9)

Kladné směry os x a y označujeme na koncích jejich znázorněných obrazů šipkami (tzn. každá z obou os pokračuje v zobrazeném směru nekonečně daleko.).



Obrázek 13: Zobrazení reálných číselných os x a y kartézské soustavy souřadnic

Osy x a y rozdělí dvourozměrnou rovinu na čtyři disjunktní pravé úhly (splývají vždy pouze styčná ramena příslušných vedlejších úhlů), které nazýváme kvadranty. Tyto kvadranty označujeme římskými číslicemi I, II, III a IV. Počátek O souřadnicového systému je společným vrcholem všech těchto čtyř pravých úhlů. ([2], str. 8)



Obrázek 14: Ukázka rozdělení dvourozměrné roviny na čtyři kvadranty

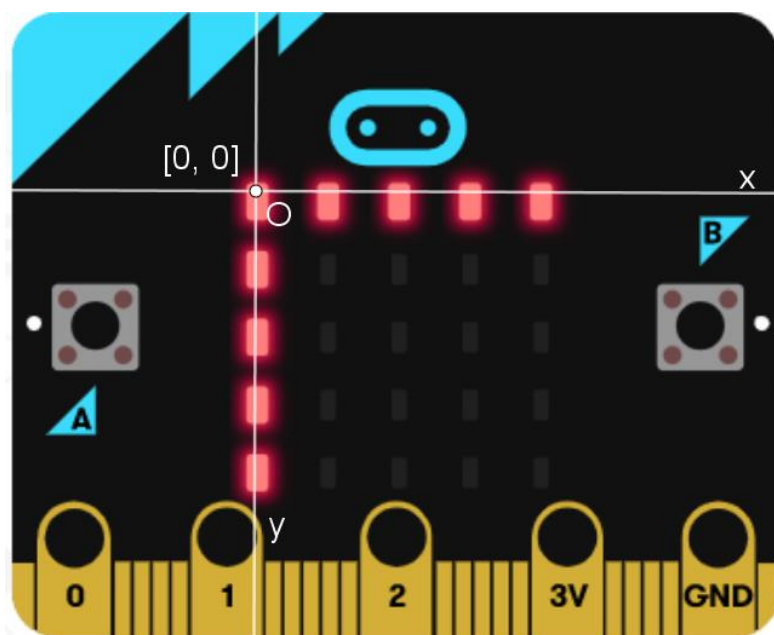
Jednotka měření

Počátku O přiřadíme v uvedeném kartézském souřadnicovém systému číslo 0 a libovolnému bodu E na ose x přiřadíme číslo 1. Délka úsečky OE je jednotkou měření na ose x , tj. $|OE| = 1$. ([2]; str. 6)

Pak každému bodu A ležícímu na ose x přiřadíme jediné reálné číslo a_1 (tj. x -ovou souřadnici bodu A) tak, že velikost úsečky OA je rovna a_1 . Přitom reálné číslo a_1 je rovno n násobku, kde $n \in \mathbb{R}$, délky úsečky OE . Pokud bod A leží na ose x napravo od počátku O , tj. leží na kladné části osy x , je číslo a_1 kladné. Pokud leží bod A na ose x nalevo od počátku O , tj. leží na záporné části osy x , pak je číslo a_1 záporné. ([2]; str. 6)

Souřadnicový systém v micro:bitu

Počátek kartézského souřadnicového systému v rovině bývá umístěn zpravidla tak, že kladný směr osy x směřuje od něj vodorovně doprava a kladný směr osy y je od něj orientován svisle nahoru. Počátek souřadnicového systému u micro:bitu splývá s diodou umístěnou v levém horním rohu matice LED diod. Kladný směr osy x směřuje od něj vodorovně doprava a kladný směr osy y je od něj naopak orientován směrem svisle dolů.



Obrázek 15: Ukázka zobrazení souřadnicového systému v micro:bitu

Jako jednu jednotku budeme v micro:bitu uvažovat vzdálenost mezi dvěma sousedními LED diodami ve vodorovném či ve svislém směru.

1.3.2 Body

Bod je jedním z tzv. primitivních pojmů geometrie. Tj. jedná se o základní pojem, který nedefinujeme.

Ke každému bodu A v rovině s danou kartézskou soustavou souřadnic přiřadíme jeho souřadnice takto:

- Bodem A vedeme kolmici k ose x a průsečík této kolmice a osy x označíme A_1 . Bod A_1 má na ose x souřadnici a_1 .
- Bodem A vedeme kolmici k ose y a průsečík této kolmice a osy y označíme A_2 . Bod A_2 má na ose y souřadnici a_2 . ([2], str. 9)

Souřadnice bodu A v kartézské soustavě souřadnic představuje uspořádaná dvojice reálných čísel a_1 a a_2 , píšeme $A [a_1, a_2]$. ([2], str. 9)

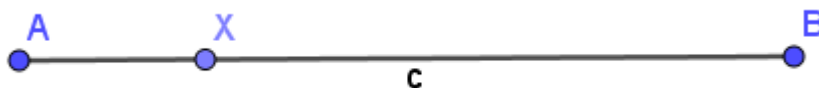
Podle znamének u jednotlivých souřadnic bodů lze bez zobrazení bodů v soustavě souřadnic určit, ve kterém kvadrantu příslušný bod leží. ([9], str. 152)

I. kvadrant	II. kvadrant	III. kvadrant	IV. kvadrant
$x > 0, y > 0$	$x < 0, y > 0$	$x < 0, y < 0$	$x > 0, y < 0$

([9], str. 152)

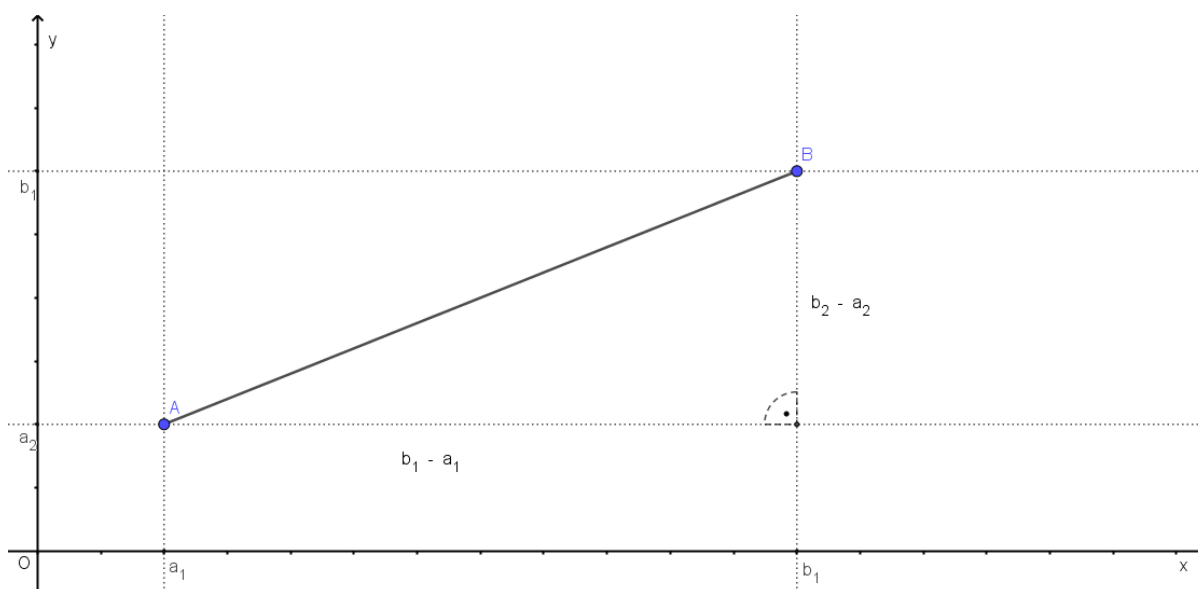
1.3.3 Úsečky a vektory

Úsečka je část přímky ohraničená dvěma jejími body, např. A a B . Určující body A a B jsou krajními body úsečky c . Úsečku c lze značit AB nebo BA pomocí jejích krajních bodů A, B . Vnitřními body úsečky $c \equiv AB$ rozumíme všechny body X , které leží mezi body A, B . ([1], str. 146)



Obrázek 16: Úsečka AB s jejím vnitřním bodem X

Velikost úsečky, nebo také vzdálenost mezi body $A [a_1, a_2]$ a $B [b_1, b_2]$ počítáme s využitím Pythagorovy věty: $|AB| = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2}$.



Obrázek 17: Zobrazení úsečky AB v kartézské soustavě souřadnic

Vektorem rozumíme množinu všech shodně orientovaných, stejně dlouhých a navzájem rovnoběžných úseček. ([2], str. 13) Každou z těchto úseček nazýváme umístěním vektoru.

Dva vektory u a v nazveme (ne)souhlasně orientované, jsou-li jejich libovolná umístění (ne)souhlasně orientované úsečky.

Je nutné rozlišovat geometrickou a aritmetickou reprezentaci vektorů.

Geometrická reprezentace vektoru

Geometrickou reprezentací vektoru rozumíme znázornění jeho libovolného umístění v kartézské soustavě souřadnic.

Aritmetická reprezentace vektoru

Aritmetickou reprezentací vektoru chápeme vyjádření vektoru pomocí jeho souřadnic. Pro zavedení souřadnic vektoru uvažujeme kartézskou soustavu souřadnic s počátkem O_{xy} a s navzájem kolmými reálnými osami x a y . Na osách x a y můžeme po řadě definovat dvojici jednotkových vektorů $\mathbf{i} = (1, 0)$ a $\mathbf{j} = (0, 1)$. Jednotkové vektory přitom volíme tak, aby jejich umístění měla společný počátek v počátku O_{xy} kartézského souřadnicového systému a aby se jejich směry shodovaly s kladnými směry souřadnicových os.

Skupinu vektorů $\mathbf{u}, \mathbf{v}, \dots, \mathbf{z}$ nazveme lineárně nezávislou, jestliže pro její lineární kombinaci $\alpha_1 \cdot \mathbf{u} + \alpha_2 \cdot \mathbf{v} + \dots + \alpha_n \cdot \mathbf{z}$, kde $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbf{R}$, platí, že $\alpha_1 \cdot \mathbf{u} + \alpha_2 \cdot \mathbf{v} + \dots + \alpha_n \cdot \mathbf{z} = \mathbf{0} = (0, 0)$ pouze, když $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$.

Je možné dokázat, že dvojice jednotkových vektorů $\mathbf{i}$ a $\mathbf{j}$ je lineárně nezávislá, proto lze libovolný vektor $\mathbf{u}$ vyjádřit jako jejich lineární kombinaci, tj. zapsat jej ve tvaru $\mathbf{u} = u_1 \mathbf{i} + u_2 \mathbf{j}$, kde $u_1, u_2 \in \mathbf{R}$. Vektory $u_1 \mathbf{i}, u_2 \mathbf{j}$ nazýváme složkami vektoru $\mathbf{u}$ a reálná čísla u_1, u_2 souřadnicemi vektoru $\mathbf{u}$. Vektor $\mathbf{u}$ pak píšeme ve tvaru $\mathbf{u} = (u_1, u_2)$.

Souřadnice vektoru $\mathbf{u}$ je ale také možné vyjádřit pomocí souřadnic jeho krajních bodů. Nechť je ve dvourozměrné rovině dán libovolný bod K . Vektor $\mathbf{OK}$ lze vyjádřit jako lineární kombinaci vektorů $\mathbf{i}, \mathbf{j}$, tj. $\mathbf{OK} = x_K \mathbf{i} + y_K \mathbf{j}$, přitom čísla x_K, y_K nazveme (kartézskými) souřadnicemi bodu K a píšeme $K = [x_K, y_K]$. Nechť je v téže dvourozměrné rovině dále dán bod $L = [x_L, y_L]$. Je-li $\mathbf{LK}$ umístění vektoru $\mathbf{u}$, pak souřadnice vektoru $\mathbf{u}$ určíme ze vztahu $\mathbf{OK} + \mathbf{u} = \mathbf{OL}$, tj. $\mathbf{u} = \mathbf{OL} - \mathbf{OK} = x_L \mathbf{i} + y_L \mathbf{j} - x_K \mathbf{i} - y_K \mathbf{j} = (x_L - x_K) \mathbf{i} + (y_L - y_K) \mathbf{j}$, odkud $\mathbf{u} = (x_L - x_K, y_L - y_K) = K - L = (u_1, u_2)$. Přitom u_1, u_2 značí souřadnice vektoru $\mathbf{u}$. ([2], str. 14)

Velikost vektoru $\mathbf{u}$ je velikost libovolného umístění orientované úsečky KL . Určení této velikosti opět vyplývá z aplikace Pythagorovy věty: $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2}$. ([2], str. 15)

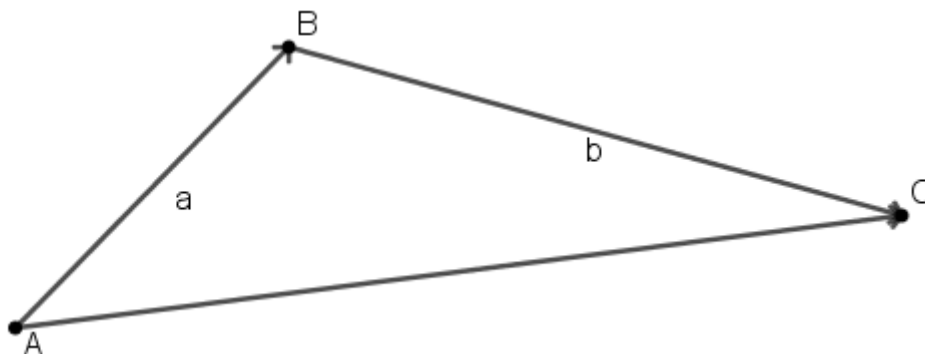
Velikost vektoru $\mathbf{u}$ nemůže být rovna libovolnému reálnému číslu. Velikost každého vektoru $\mathbf{u}$ je rovna nezápornému číslu, tzn. $|\mathbf{u}| \geq 0$. Velikost vektoru $\mathbf{u}$ je rovna nule právě tehdy, když je tento vektor nulový. Tj. $|\mathbf{u}| = 0 \Leftrightarrow \mathbf{u} = (0, 0)$. ([2], str. 15)

S vektory lze provádět podobné operace jako s reálnými čísly. Vektory lze např. sčítat, odčítat a násobit reálným číslem.

Grafické sčítání geometrické reprezentace vektorů

V rovině jsou dány dva nekolineární vektory $\mathbf{a}, \mathbf{b}$ (*Poznámka:* Vektory rovnoběžné se stejnou přímkou nazýváme kolineární. Tzn. dva vektory, které nejsou rovnoběžné

s jednou a touž přímkou, nazveme nekolineární.) a bod A . Do něj umístíme vektor $\mathbf{a}$ jeho počátečním bodem, koncový bod tohoto umístění vektoru $\mathbf{a}$ označme B . Do bodu B umístíme vektor $\mathbf{b}$ jeho počátečním bodem a koncový bod tohoto umístění vektoru $\mathbf{b}$ označme C . Vektor $\mathbf{AC}$ pak nazveme grafickým součtem vektorů $\mathbf{a}$, $\mathbf{b}$. ([2], str. 16)



Obrázek 18: Grafické sčítání vektorů

Sčítání aritmetických vektorů

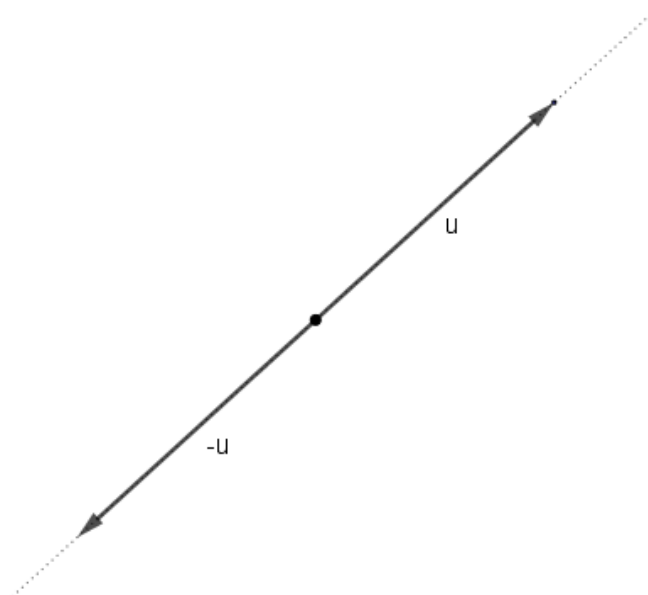
Sčítání aritmetických vektorů, tj. vektorů zapsaných pomocí jejich souřadnic, má podobné vlastnosti jako sčítání reálných čísel. Pro operaci sčítání aritmetických vektorů platí tedy pro libovolné tři různé nekolineární vektory $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ téže roviny následující vlastnosti:

- komutativita: $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$,
- asociativita: $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$,
- existence neutrálního prvku: $\mathbf{u} + \mathbf{0} = \mathbf{0} + \mathbf{u} = \mathbf{u}$, kde $\mathbf{0}$ je tzv. nulový vektor. ([2], str. 17)

Opačný vektor

Z reálných čísel známe pojem opačné číslo. U vektorů definujeme podobný pojem, a to pojem opačný vektor. Opačným vektorem přitom rozumíme takový vektor, který je s daným vektorem kolineární, tj. oba vektory jsou rovnoběžné s touž přímkou, má stejnou velikost jako daný vektor, ale jeho orientace je oproti danému vektoru opačná. Mějme vektor $\mathbf{KL}$ určený dvojicí různých bodů K, L roviny, pak vektor k němu opačný je vektor $\mathbf{LK}$. ([2], str. 17)

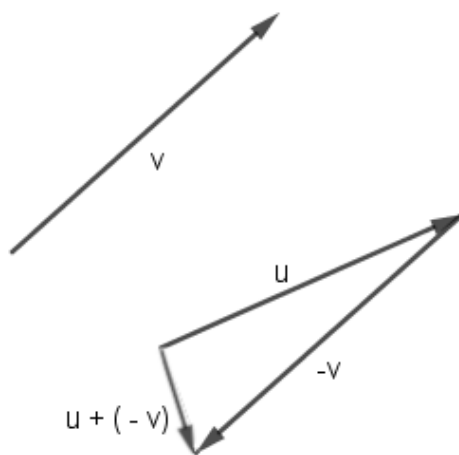
Vektor opačný k vektoru $\mathbf{u}$ označujeme $-\mathbf{u}$ a platí: $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.



Obrázek 19: Znáznornění opačného vektoru k vektoru $\mathbf{u}$

Grafický rozdíl geometrické reprezentace vektorů

Mějme v rovině dány dva nekolineární vektory $\mathbf{u}$ a $\mathbf{v}$. Grafický rozdíl vektorů $\mathbf{u}$ a $\mathbf{v}$ určíme jako grafický součet vektoru $\mathbf{u}$ a vektoru $-\mathbf{v}$, který je opačným vektorem k vektoru $\mathbf{v}$, tzn. symbolicky můžeme psát $\mathbf{u} - \mathbf{v} = \mathbf{u} + (-\mathbf{v})$. ([2], str. 18)



Obrázek 20: Grafický rozdíl vektorů $\mathbf{u}$ a $\mathbf{v}$

Násobení vektoru reálným číslem

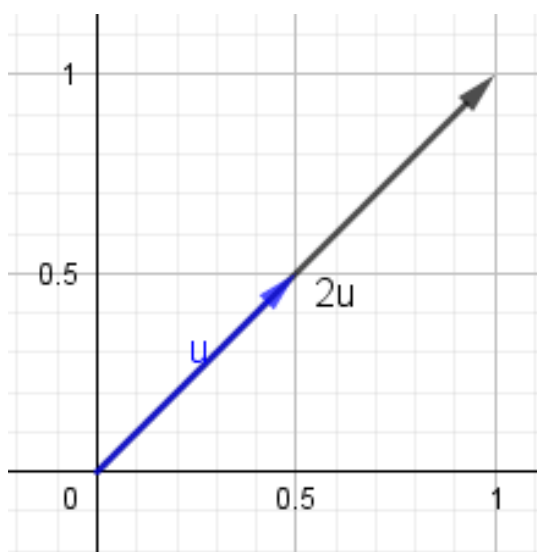
Změna velikosti vektoru se provádí násobením vektoru nezáporným reálným číslem. Násobení vektoru záporným reálným číslem chápeme jako násobení opačného vektoru absolutní hodnotou tohoto čísla. ([2], str. 19)

Vektor $\mathbf{u}$ můžeme vynásobit libovolným reálným číslem, čímž dostaneme nový vektor $\mathbf{v}$, který je s původním vektorem kolineární, ale oproti původnímu vektoru $\mathbf{u}$ má

jinou velikost. Kolineárnost vektorů $\mathbf{u}$ a $\mathbf{v}$ zapíšeme symbolicky ve tvaru $\mathbf{v} = k \cdot \mathbf{u}$, kde $k \in \mathbb{R}$.

Pro dvě libovolná reálná čísla k, l a pro dva libovolné nekolineární vektory $\mathbf{u}, \mathbf{v}$ platí následující vlastnosti:

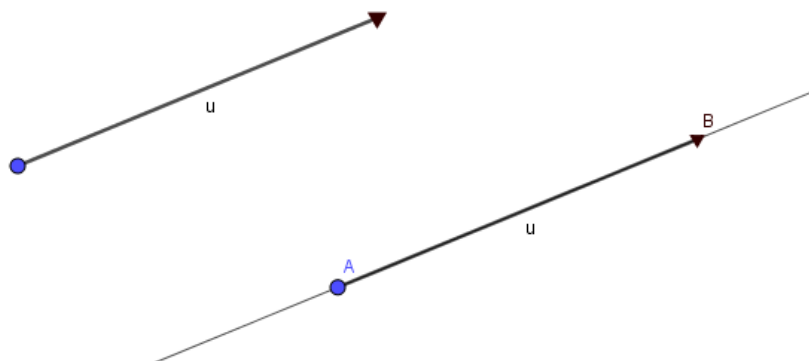
- asociativita: $k \cdot (l \cdot \mathbf{u}) = (k \cdot l) \cdot \mathbf{u}$,
- distributivita:
 - $k \cdot (\mathbf{u} + \mathbf{v}) = k \cdot \mathbf{u} + k \cdot \mathbf{v}$,
 - $(k + l) \cdot \mathbf{u} = k \cdot \mathbf{u} + l \cdot \mathbf{u}$.



Obrázek 21: Ukázka násobení vektoru reálným číslem

1.3.4 Přímky

V rovině jsou dány dva různé body A a B , pak vektor $\mathbf{u} = B - A$ je směrovým vektorem přímky AB . Vzhledem k tomu, že vektor má nekonečně mnoho různých umístění v rovině, nemusí nutně směrový vektor $\mathbf{u}$ přímky AB ležet na této přímce AB . Směrový vektor $\mathbf{u}$ přímky AB je s touto přímkou rovnoběžný, tzn. má s ní shodný směr. Směrový vektor sám o sobě neurčuje jednu konkrétní přímku, ale určuje všechny přímky roviny, které náležejí jeho směru. K jednoznačnému určení jedné konkrétní přímky v rovině je potřeba alespoň jeden bod ležící na této přímce. ([4], str. 66) ([2], str. 27)



Obrázek 22: Zobrazení přímky AB a dvou zvolených reprezentantů jejího směrového vektoru $\mathbf{u}$

Pomocí směrového vektoru $\mathbf{u}$ přímky a daného bodu A ležícího na přímce, lze jednoznačně určit konkrétní přímku, označme ji například p . Tuto jednoznačnou určenost přímky p je možné zapsat s využitím souřadnic a_1, a_2 daného bodu A a souřadnic u_1, u_2 směrového vektoru $\mathbf{u}$, a to pomocí tzv. parametrických rovnic

$$p: x = a_1 + t \cdot u_1,$$

$$y = a_2 + t \cdot u_2,$$

kde t je parametr, $t \in \mathbb{R}$, a kde x, y jsou souřadnice libovolného bodu X přímky p . ([2], str. 28)

Pokud parametr $t \in \langle 0; \infty \rangle$, pak se jedná o polopřímku, a pokud by parametr t náležel např. intervalu $\langle 0, 1 \rangle$, pak by daným geometrickým objektem byla úsečka. ([2], str. 29)

Parametrické vyjádření přímky ležící v rovině lze pomocí řešení soustavy lineárních algebraických rovnic s jedním parametrem t převést na obecnou rovnici přímky. Při převodu se upravuje soustava lineárních algebraických rovnic na tvar bez parametru t , tzn. při převodu je třeba reálný parametr t z rovnic vyeliminovat. V závěru je třeba upravit výslednou obecnou rovnici přímky tak, aby levá strana obsahovala součet (v tomto pořadí) proměnné x násobené jejím koeficientem, proměnné y násobené jejím koeficientem a reálné konstanty a aby na pravé straně rovnice bylo číslo 0. ([2], str. 31)

Z obecné rovnice přímky p lze početními ekvivalentními úpravami získat směrnice tvar rovnice přímky, tj. tvar $y = k \cdot x + q$, kde $k = \tan(\alpha)$ je směrnice přímky p , α je orientovaný úhel mezi kladnou poloosou x a přímkou p , a to takovým způsobem, že z obecné rovnice přímky p vyjádříme proměnnou y (tj. y -ovou souřadnici libovolného bodu X přímky p) v závislosti na proměnné x (tj. na x -ové souřadnici libovolného bodu X přímky p). ([2], str. 33)

2 Programování

V této kapitole jsou představeny programovací jazyky, pomocí nichž lze programovat micro:bity v online vývojovém prostředí „makecode.microbit.org“. V poslední části této kapitoly jsou uvedena a popsána témata z programování, která se využívají při řešení úloh ve vytvořených pracovních listech.

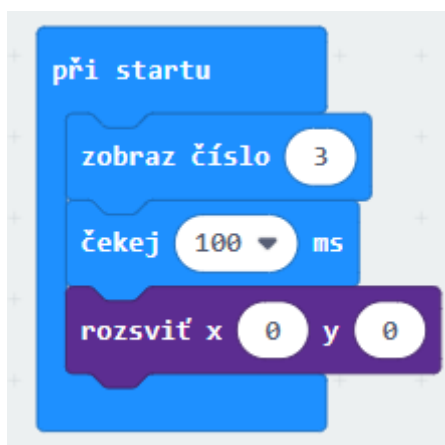
2.1 Programovací jazyk

Programovací jazyk je jazyk, který slouží k tvorbě počítačového programu. Programováním se rozumí proces, kterým zadáváme do počítače nějaký konkrétní algoritmus, tj. postup, který vede k řešení nějakého problému. [IV]

2.2 Blokově orientované programování

Většina blokově orientovaných jazyků, které najdeme ve webovém prohlížeči, jako je Scratch nebo micro:bit - makecode, využívá JavaScript knihovnu zvanou Blockly. Ta prezentuje příkazy jako bloky. Tato knihovna také umožňuje přepínání do jiných jazyků, jako jsou např. JavaScript nebo Python. To znamená, že kód vytvořený pomocí bloků se automaticky přepíše do zvoleného jazyka. [III]

Jedná se o open-source knihovnu, která je přeložena do více jak 40 jazyků. [III]



Obrázek 23: Ukázka blokově orientovaného jazyka micro:bit – makecode

2.3 JavaScript

JavaScript je programovací jazyk určený pro psaní scriptů (tj. krátkých programů, které reagují na dění na webové stránce). Můžeme tedy říct, že se jedná o jazyk, který doplňuje statický jazyk HTML (který není programovacím jazykem, ale „značkovacím jazykem“, tj.

jazykem sloužícím pro popis obsahu webové stránky). JavaScript popisuje chování webové stránky. ([9], str. 15)

Často dochází k záměně s programovacím jazykem Java. Tyto jazyky se ale zásadně liší. JavaScript se využívá k programování jednoduchých programů, protože se tyto programy vykonávají pomaleji než v Javě. Tu můžeme využít k programování složitějších a komplexnějších programů. ([9], str. 16)

2.4 Python

Jedná se o moderní programovací jazyk, který umožňuje poměrně snadným způsobem navrhovat jednoduché programy, ale zároveň obsahuje širokou škálu knihoven, které uživatelům nabízí nejrůznější pomocné podprogramy. Uživatel tedy nemusí sám programovat různé drobné podprogramy, např. třídění, generování náhodného čísla apod. Ty jsou již předprogramovány. ([6], str. 22)

2.5 Témata v programování

V této podkapitole jsou představena témata programování, jejichž znalost je nezbytná pro úspěšné řešení úloh navržených v pracovních listech.

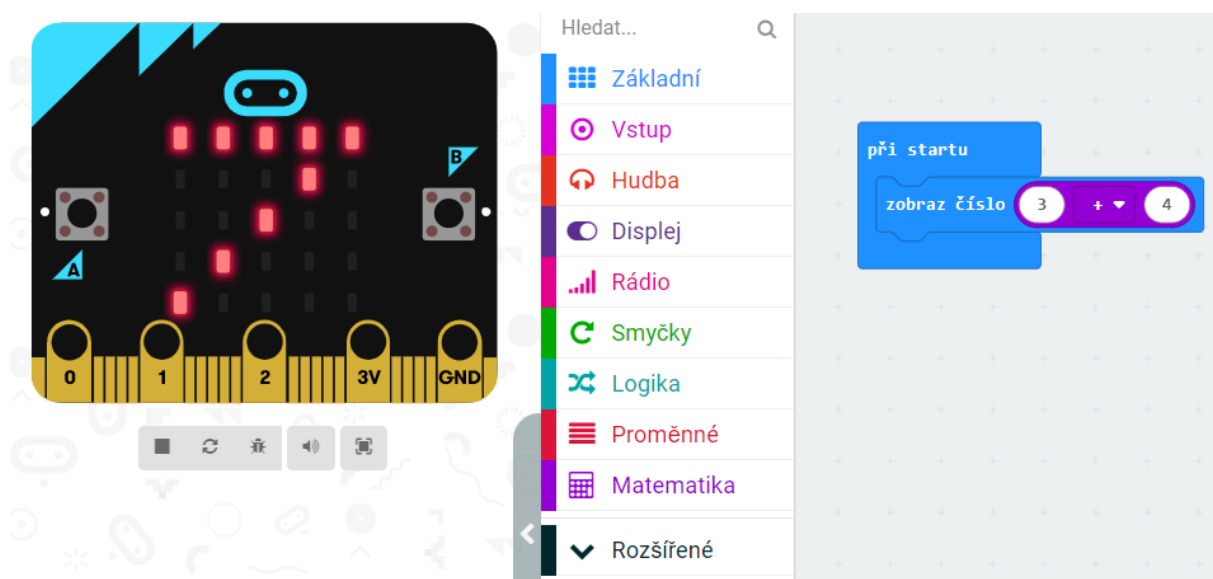
2.5.1 Základní matematické operace a vypisování

Blokově orientované programování

V blokově orientovaném programování micro:bitu najdeme základní matematické operace v záložce „Matematika“. Je důležité bloky umísťovat především dle jejich tvaru.

V makecode nefunguje žádná konzole pro výpis výsledků, ty nám vypisuje přímo micro:bit (pro podrobnosti viz str. 51). K vypsání výsledku je tedy potřeba využít i nějaký základní blok ze složky „Základní“, např. „Zobraz číslo ___“ nebo „Zobraz text ___“.

Aby kód vypsál výsledek, je také nutné jej umísťovat do bloků „Začátek programu“ jako jsou např. „při startu“, „opakuj stále“, „po stisknutí tlačítka ___“ a jiné.



Obrázek 24: Blokově orientované programování – základní matematická operace (sčítání)

JavaScript a Python

Výpočty v JavaScript a Pythonu jsou v podstatě stejné. Pro zápisy výpočtů se užívá tzv. infixového zápisu, tzn. operátor je mezi čísly.

Pro programování v obou těchto jazycích je potřeba knihovna pro micro:bit a podobně, jako pro blokově orientované programování, i zde platí, že se zde nenachází žádná konzole, a je tedy nutné výsledek ukazovat přímo na micro:bitu.

Nejprve je třeba říci, o jaký typ předdefinované funkce se jedná. V tomto případě se jedná o předdefinované funkce „basic“ a pro výpis čísla pak máme funkci „Shownumber(3+4)“.

Zápis v konkrétních jazycích pak vypadá takto:

JavaScript	Python
<code>basic.showNumber(3 + 4)</code>	<code>basic.show_number(3 + 4)</code>

2.5.2 Proměnná a základní datové typy

Proměnná je „základní stavební jednotka“ ve většině programovacích jazyků, do které je možné uložit informace (data) potřebné pro běh programu. Do proměnné lze ukládat různé druhy dat. Tyto druhy nazýváme datové typy. [V]

Datové typy

Při programování micro:bitu se můžeme setkat s několika základními datovými typy: text, znak, celé číslo, desetinné číslo, boolean,

Datové typy rozdělujeme na dva druhy:

- jednoduché
 - Celá čísla – jedná se o typ zvaný `integer`; ve většině jazyků mají celá čísla omezený rozsah, a to od $-32\,768$ do $32\,767$
 - Desetinná čísla (reálná čísla) – možná označení `double`, `float` nebo `real`; rozsah se dle typu liší, pro `double` to je např. $1.7 \cdot 10^{-308}$ až $1.7 \cdot 10^{308}$
 - Znak – jedná se o typ zvaný `char`; znak je v počítači reprezentován pomocí celého čísla, většinou se používá znaková sada ASCII
 - Boolean – nabývá hodnot `true` (pravda, 1) nebo `false` (nepravda, 0)
- složené
 - Textový řetězec – `string`; posloupnost po sobě jdoucích znaků
 - Pole – zvané v JavaScriptu a Pythonu jako `array`; jedná se o množinu prvků stejného datového typu pole, může být vícerozměrné (pak se označuje jako matice)
 - Seznam – označovaný jako `list`; obdoba pole, ale lze do něj vkládat různé datové typy

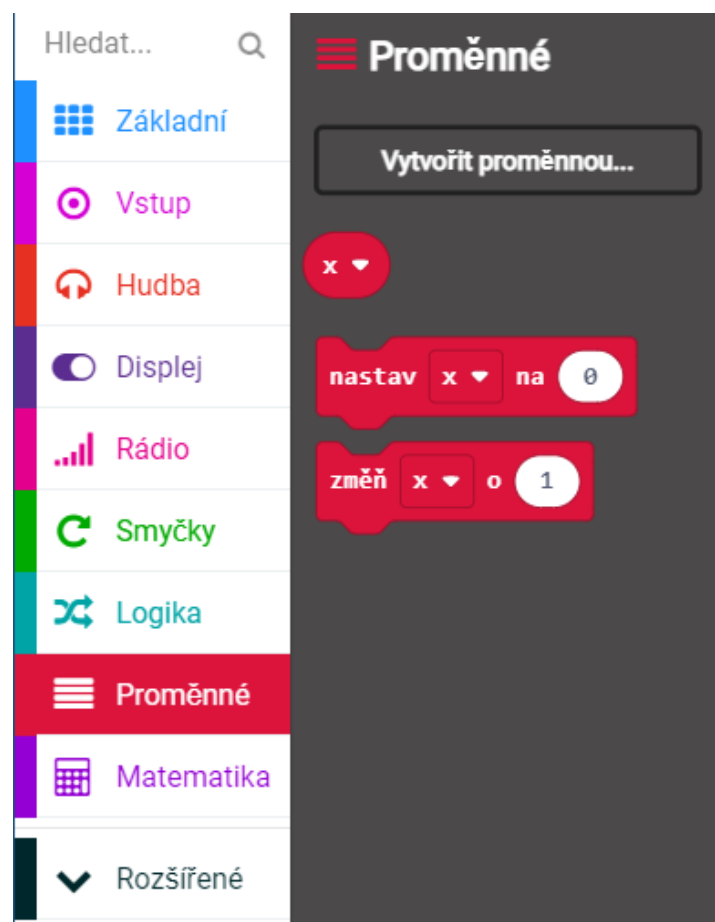
Složené datové typy vznikají skládáním jednoduchých typů.

Obecně lze datové typy rozdělit na homogenní a heterogenní. Homogenní datové typy obsahují jeden nebo více prvků stejného datového typu, např. textový řetězec, nebo pole. Heterogenní datové typy jsou složeny z prvků různých datových typů. Některými z představitelů heterogenního datového typu jsou například seznamy v jazyce Scheme, nebo typ `struct` v jazyce C.

Související pojmy s proměnnou a datovými typy jsou deklarace a inicializace. Deklarace proměnné je nastavení datového typu, např. `int x` (v jazyce C), což představuje, že `x` je celé číslo. Inicializace je přiřazení hodnoty proměnné, např. `x = 7`. Inicializace probíhá po deklaraci, případně deklarace a inicializace mohou probíhat najednou, např. `int x = 7`.

Blokově orientované programování

V blokově orientovaném jazyce `makecode` je pro vytvoření proměnné definována záložka „Proměnné“. Tato záložka obsahuje příkazy pro vytvoření proměnné a poté jsou zde bloky pro práci s nimi.



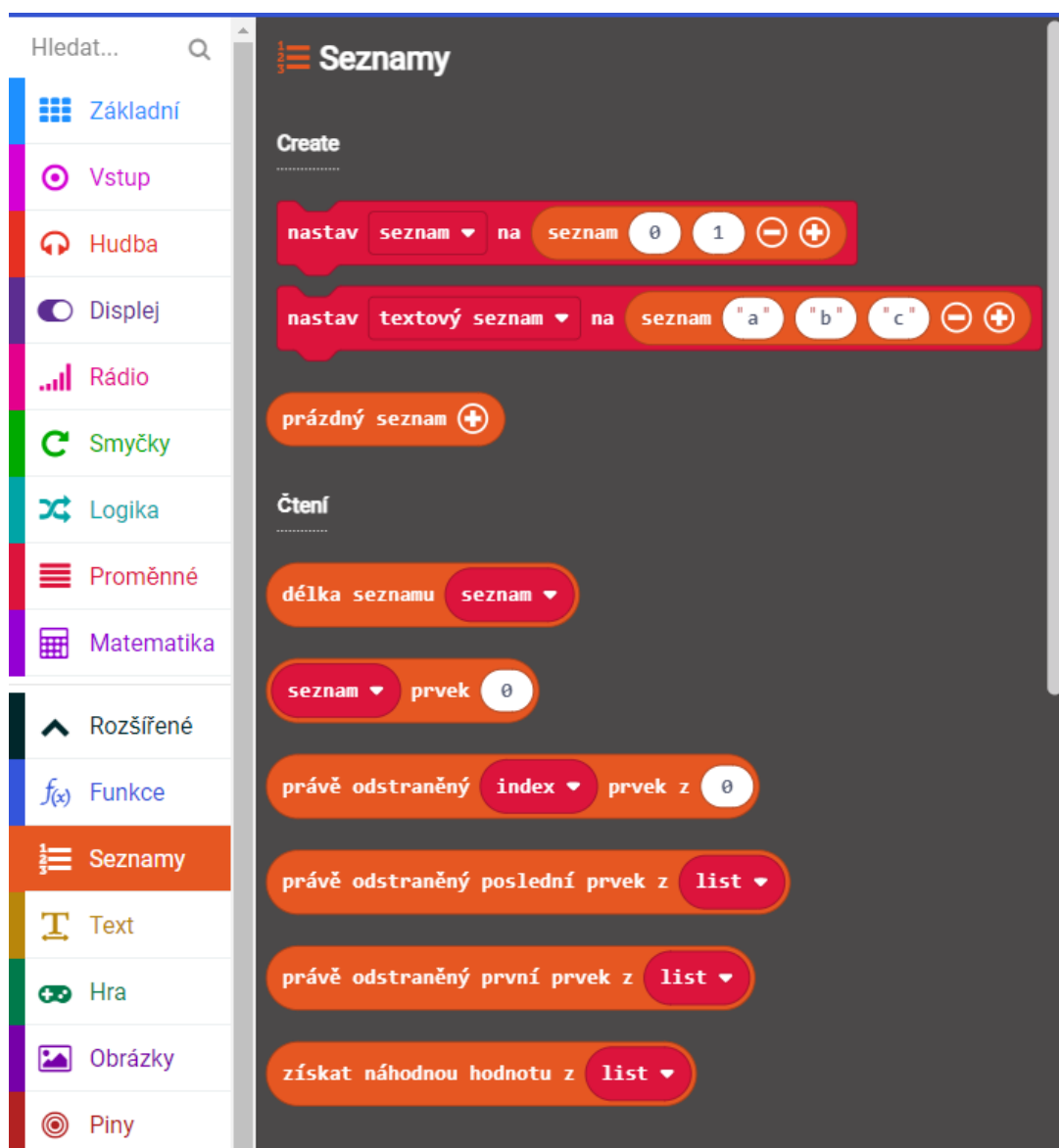
Obrázek 25: Blokově orientované programování – záložka „Proměnné“

Datové typy v blokově orientovaném programování rozdělujeme na čísla a na text. Pokud chceme vypisovat text nebo číslo, je třeba zvolit vhodný blok.



Obrázek 26: Blokově orientované programování – textové a číselné datové typy

V blokově orientovaném programování se také nachází jistá alternativa pole. Jedná se o seznam, který se však chová jako pole. V těchto seznamech musí být prvky pouze stejného datového typu.



Obrázek 27: Seznamy v blokově orientovaném programování

JavaScript a Python

Konstrukce proměnné je v těchto dvou jazycích velmi podobná.

Pro deklaraci proměnné v jazyce JavaScript se používá slovo `let`. V jazyce Python probíhá inicializace společně s deklarací. Pro změnu proměnné se v obou jazycích používá výraz `change`.

JavaScript	Python
<code>let item: number</code>	
<code>item = 0</code>	<code>item = 0</code>

Poznámka: Výraz `item` je název proměnné.

2.5.3 Podmínky a logika

Podmínky slouží v programování k větvení programu, tj. k rozdělení programu na více částí, které mají za určitých podmínek vykonávat jiné příkazy.

S podmínkami souvisí logika a vyhodnocování pravdivostí výroků. Podmínka může být vyhodnocena vždy buď jako `true`, nebo `false`. Pokud je podmínka vyhodnocena jako `true`, posloupnost příkazů v podmínce proběhne, pokud je naopak vyhodnocena jako `false`, pak posloupnost příkazů v podmínce neproběhne.

Do podmínek se vkládají pravdivostní výroky, pro určování pravdivostních výroků se používají relační operátory.

Porovnávat můžeme vždy pravdivostní výroky vytvořené pouze pro stejné datové typy.

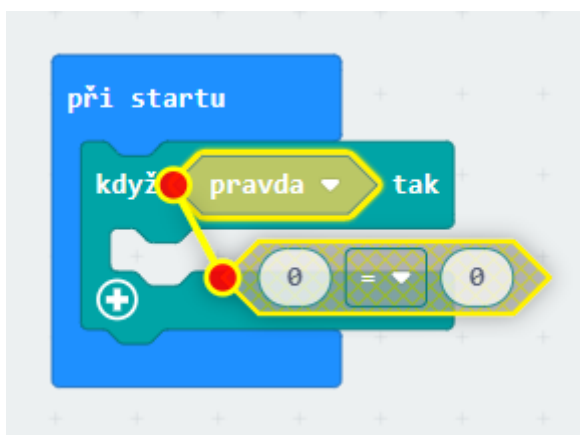
V rámci podmínek existují logické funkce, které umožňují tvorbu disjunkce, konjunkce a negace. Disjunkce bývá v programování označována jako `OR`, konjunkce jako `AND` a negace se označuje `NOT`.

Blokově orientované programování

V blokově orientovaném programování existují dva typy podmíněných příkazů:

- Když <podmínka> tak ____
- Když <podmínka> tak ____ jinak ____

Relační i logické operátory jsou v blokově orientovaném jazyce definovány opět pomocí bloků, tyto bloky mají tvar šestiúhelníků, které se vkládají do míst jim předurčených. Viz obr. 28.



Obrázek 28: Ukázka podmíněného příkazu v blokově orientovaném programování

V blokově orientovaném programování lze kromě porovnávání čísel také porovnávat textové řetězce.



Obrázek 29: Relační operátory v blokově orientovaném programování

Logické operátory AND, OR a NOT jsou zadávány opět pomocí šestiúhelníků.



Obrázek 30: Logické operátory v blokově orientovaném programování

JavaScript a Python

Zápisy relačních operátorů jsou pro JavaScript a Python stejné:

	JavaScript	Python
Rovnost:		==
Je ostře větší:		>
Je ostře menší:		<
Je větší nebo rovno:		>=
Je menší nebo rovno:		<=
Nerovnost:		!=

Označení logických operátorů se v JavaScript a Pythonu liší:

	JavaScript	Python
a zároveň	& &	And
nebo		Or
negace	!	Not

V obou uvedených jazycích je podmínka označena klíčovým slovem `if` a také je možná varianta `if else`, avšak syntaxe se odlišuje:

	JavaScript	Python
If	<code>if (podmínka){ pokud je podmínka splněna, proběhne tento kód;}</code>	<code>if podmínka: pokud je podmínka splněna, proběhne tento kód</code>
If else	<code>if (podmínka){ pokud je podmínka splněna, proběhne tento kód;}else{ pokud není podmínka splněna, proběhne tento kód;}</code>	<code>if podmínka: pokud je podmínka splněna, proběhne tento kód else: pokud není podmínka splněna, proběhne tento kód</code>
Více podmíněných příkazů za sebou	<code>if (podmínka1){ pokud je podmínka1 splněna, proběhne tento kód;} else if (podmínka2){ pokud je podmínka2 splněna, proběhne tento kód; } else { pokud není podmínka1 i podmínka2 splněna, proběhne tento kód;}</code>	<code>if podmínka1: pokud je podmínka1 splněna, proběhne tento kód elif podmínka2: pokud je podmínka2 splněna, proběhne tento kód else: pokud není podmínka1 i podmínka2 splněna, proběhne tento kód</code>

V jazyce JavaScript existuje ještě jeden typ větvení, a to větvení pomocí příkazu `switch`, pomocí kterého dochází k větvení podle různých hodnot proměnné.

2.5.4 Cykly

Cykly nebo také smyčka, či anglicky loop slouží v programování pro opakování nějakého konkrétního kódu. Dle daných vstupních podmínek můžeme obecně cykly rozdělit na tři základní typy: cyklus se známým počtem opakování, cyklus s podmínkou na začátku či cyklus s podmínkou na konci.

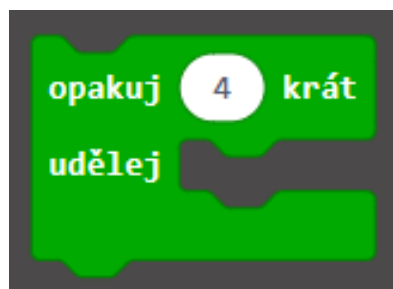
V určitých programovacích jazycích existují speciální cykly, jako například v programovacím jazyce makecode se vyskytuje nekonečněkrát opakující se cyklus. Speciální cykly se nacházejí i u programování micro:bitu, zde jsou definovány výše zmíněný nekonečný cyklus a opakování každých n ms (mikrosekund), kde $n \in \mathbf{R}^+$.

Opakování v cyklech se nazývá iterace. Cykly se často využívají k procházení polí.

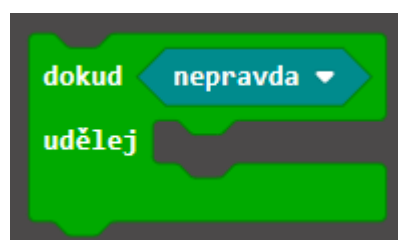
Blokově orientované programování

V blokově orientovaném programování se vyskytují klasické cykly jako cyklus s podmínkou na začátku a cyklus se známým počtem opakování. Na rozdíl od jiných programovacích jazyků zde není cyklus s podmínkou na konci. Avšak jsou zde obsaženy i cykly speciální jako nekonečný cyklus a opakování po určitém čase.

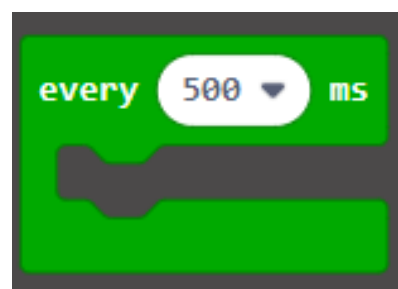
Opakování se známým počtem
kroků



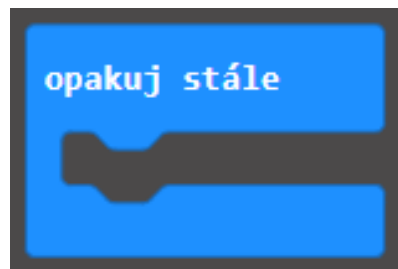
Opakování s podmínkou na
začátku



Opakování po určitém čase



Nekonečný cyklus



Obrázek 31: Ukázky bloků s cykly v blokově orientovaném programování

JavaScript a Python

JavaScript a Python mají velmi podobnou konstrukci dvou cyklů, a to cyklu `for`, který lze použít, pokud známe počet opakování, a cyklu `while`, který má podmínku na začátku.

	JavaScript	Python
Cyklus se známým počtem opakování <code>for</code>	<pre>for (inicializace proměnné, podmínka, aktualizace) { kód }</pre>	<pre>for proměnná in interval: kód</pre>
Cyklus s podmínkou na začátku <code>while</code>	<pre>while (podmínka) { kód, který se opakuje, dokud není splněna podmínka; aktualizace }</pre>	<pre>while podmínka: kód, který se opakuje, dokud není splněna podmínka</pre>

Pro předčasné ukončení cyklu lze v obou případech využít příkazu `break`. Jeho aktivací se okamžitě přestanou vykonávat příkazy v cyklu a program pokračuje vykonáváním příkazů uvedených za cyklem.

JavaScript má na rozdíl od Pythonu cyklus `do while`, jedná se o cyklus s podmínkou na konci.

	JavaScript
Cyklus s podmínkou na konci <code>do while</code>	<pre>do { kód, který se bude vykonávat, dokud bude splněna</pre>

```

    podmínka, ale na rozdíl od
    while proběhne alespoň jednou;
    aktualizace;
} while (podmínka);

```

Nekonečný cyklus lze zkonstruovat zcela jednoduše, a to tím, že v cyklu s podmínkou bude vždy taková podmínka, která bude vždy splněna.

Aktualizace je důležitá součást každého cyklu, jehož může a nemusí být součástí. Nutnou součástí aktualizace je ale cyklus se známým počtem iterací. Aktualizace mění proměnnou. Existuje několik způsobů, jak ke změně může docházet. Definujeme tedy dva operátory: inkrementace a dekrementace. Tyto operátory lze zadávat prefixově a postfixově.

Inkrementace zvětšuje hodnotu proměnné o jedna a zapisuje se ++. Dekrementace snižuje hodnotu proměnné o jedna a zapisuje se --.

Tabulka 1: Ukázka zápisu dvou operátorů – inkrementace a dekrementace

Mějme proměnnou i				
	Pre- zápis		Post- zápis	
	Prodloužený zápis:	Zkrácený zápis:	Prodloužený zápis:	Zkrácený zápis:
inkrementace	i=i+1	++i	i=1+i	i++
dekrementace	i=i-1	--i	i=1-i	i--

Jak už napovídá prodloužený zápis, záleží i na zápisu aktualizace. Postfixový i prefixový zápis zvyšuje proměnnou, ale v jiném pořadí. Prefixový zápis nejprve zvýší hodnotu a poté vrátí hodnotu, oproti tomu postfixový zápis nejprve vrátí hodnotu a poté proměnnou zvýší o jedna.

2.5.5 Funkce

V programování je funkce část kódu (nebo také podprogram), kterou lze volat a vykonat tak daný kód na místě, kde je potřeba. Funkce slouží k tomu, abychom v kódu

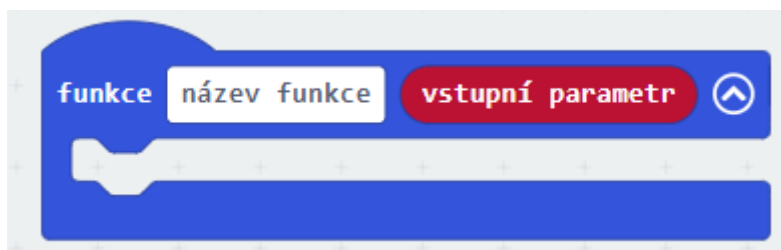
neopakovali jeho stejné části. Pomocí funkcí, větvení a cyklů zpřehledňujeme zdrojový kód. Funkce mít může, ale také nemusí mít vstupní parametry.

S pojmem funkce souvisí rekurzivní funkce. Rekurze je jev, kdy funkce volá sama sebe. Využívá se v programovacích jazycích, které nemají cykly, např. v programovacím jazyce Scheme.

Funkce může, ale nemusí vracet nějakou hodnotu. Pokud daná funkce nevrací žádnou návratovou hodnotu, pak se jedná o proceduru.

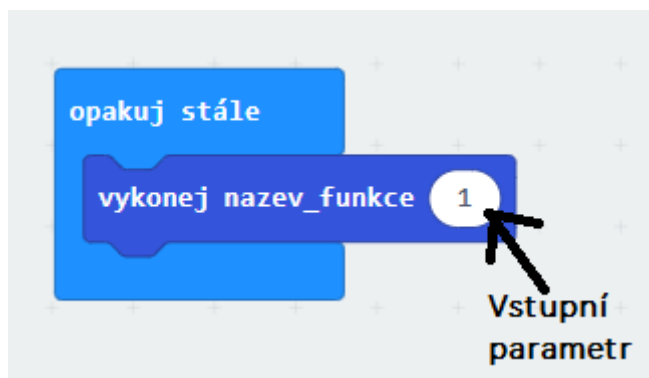
Blokově orientované programování

Funkce je v blokově orientovaném programování opět blok, který nese nějaký název a který může mít, ale nemusí mít nějaké parametry.



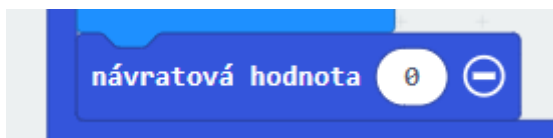
Obrázek 32: Ukázka definování funkce v blokově orientovaném programování

Volání funkce pak vypadá jako normální příkaz.



Obrázek 33: Volání funkce v blokově orientovaném programování

Funkce může, ale nemusí vracet nějakou hodnotu. V blokově orientovaném programování odpovídá funkci, která vrací na konkrétní hodnotu, ukončující blok „návratová hodnota“.



Obrázek 34: Ukončující blok návratové hodnoty v blokově orientovaném programování

Konstrukce nebo také definice funkce je v obou programovacích jazycích velmi podobná.

	JavaScript	Python
Funkce bez vstupních parametrů	<pre>function nazev_funkce () { příkazy; }</pre>	<pre>def nazev_funkce() : příkazy</pre>
Funkce se vstupními parametry	<pre>function nazev_funkce (vstupni_parametry) { příkazy; }</pre>	<pre>def nazev_funkce (vstupni_parametry) : příkazy</pre>

Pokud je potřeba více vstupních parametrů, pak se v závorkách oddělují čárkami.

Volání funkce je v obou programovacích jazycích v podstatě stejné, liší se pouze zápisem syntaxe.

JavaScript	Python
<pre>nazev_funkce(vstupni_paramet ry);</pre>	<pre>nazev_funkce(vstupni_parame try)</pre>

Pokud má funkce vrátit na nějakou návratovou hodnotu, pak k tomu slouží klíčové slovo `return`. Zápis tohoto příkazu je opět v obou jazycích v podstatě stejný a musí být umístěn uvnitř funkce, která má na danou hodnotu vrátit.

JavaScript	Python
<pre>return navratova_hodnota;</pre>	<pre>return navratova_hodnota</pre>

3 Micro:bit

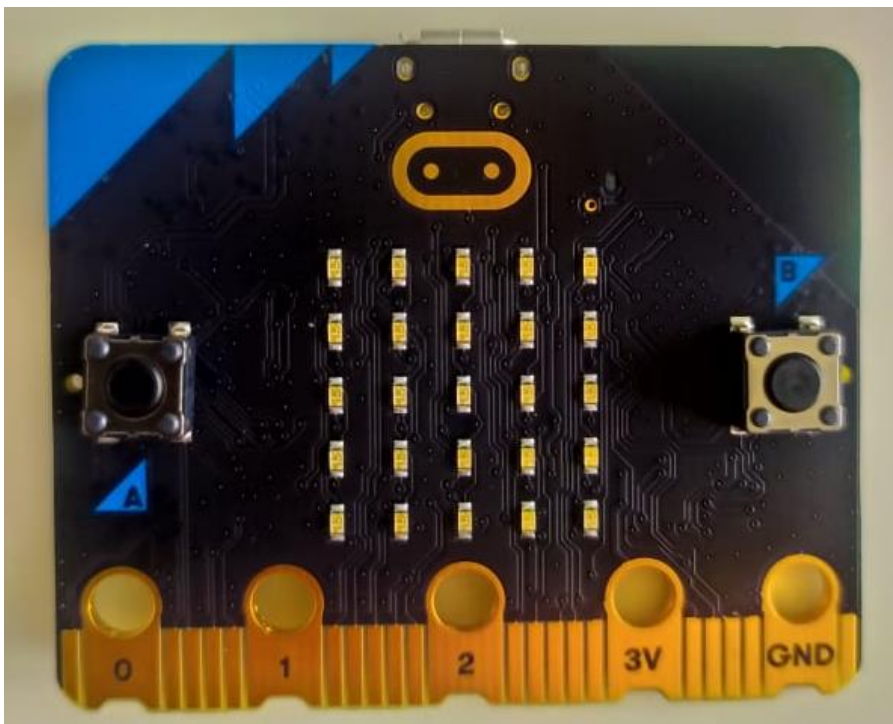
Micro:bit je programovatelný kapesní počítač vytvořený díky projektu řešenému společností BBC. Společnost BBC vyvinula tento počítač s cílem posílení digitální gramotnosti ve Spojeném království. Cílem tohoto projektu bylo posílit zájem o dovednosti ve vědě, moderních technologiích a inženýrství zejména díky tomu, že se nacházíme v digitálním věku. [I] [II]

3.1 Základní charakteristika

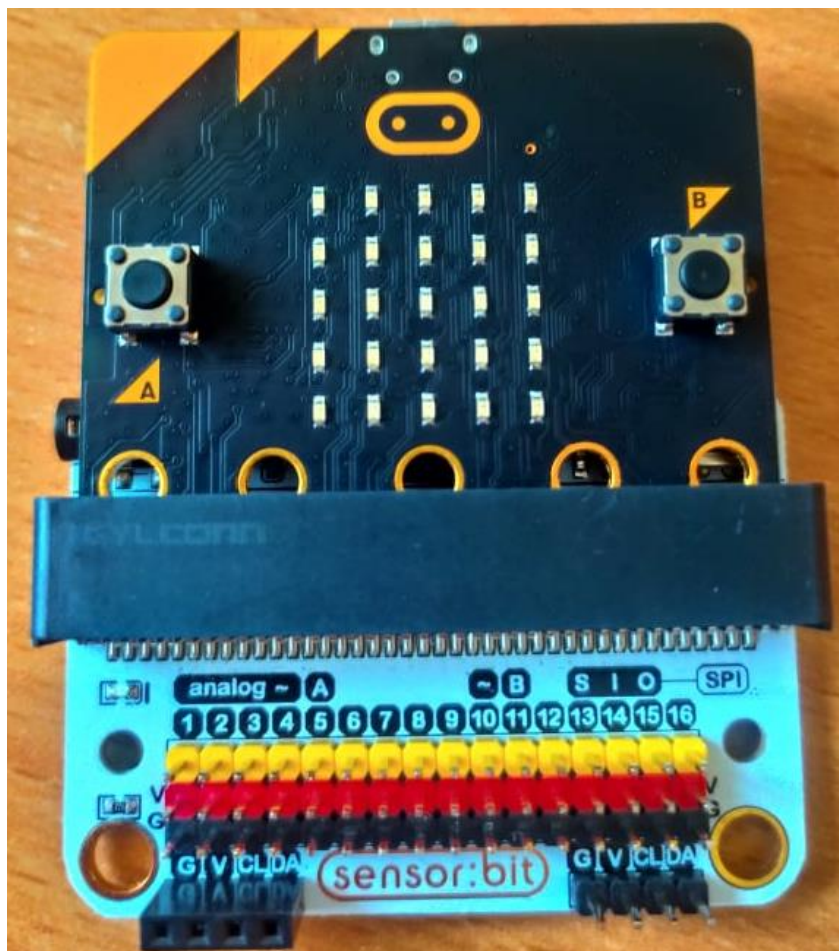
Micro:bit se skládá z matice 5 x 5 LED diod; dvou programovatelných tlačítek (označení **A** a **B**); kompasu; tříosého akcelerometru (gyroskopu); 17 GPIO pinů, přitom tři z nich jsou snadno přístupné pomocí např. krokodýlových kabelů (nebo pomocí redukce). ([5]; str. 4)

Micro:bit sám o sobě umožňuje:

- zjišťovat intenzitu osvětlení
- zjišťovat sílu magnetického pole a teplotu
- komunikaci pomocí Bluetooth
- přímou komunikaci pomocí rádia
- připojení sluchátek či reproduktoru. ([5]; str. 4 -5)



Obrázek 35: Micro:bit



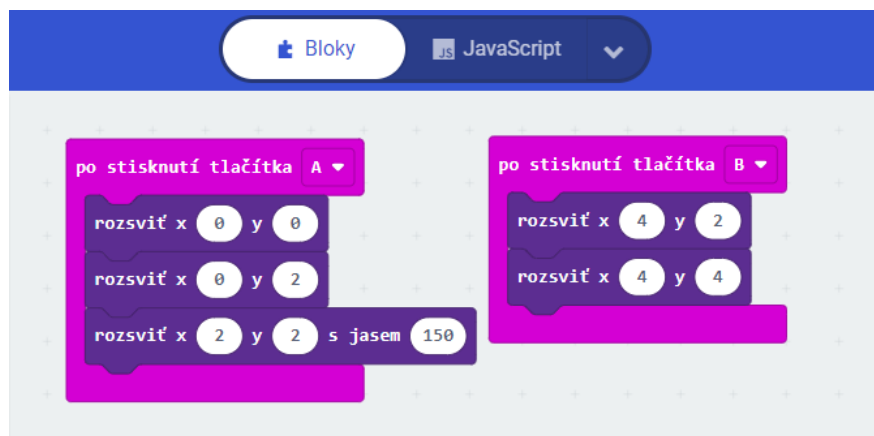
Obrázek 36: Micro:bit s redukcí

3.2 Vývojové prostředí

Micro:bit lze programovat ve velkém množství vývojových prostředí, jako je například prostředí Scratch, které využívá blokově orientované programování. Toto prostředí ovšem postrádá dobrý simulátor.

Během výuky jsem však osobně k programování micro:bitu používala prostředí <https://makecode.microbit.org/>, které má mnoho dalších funkcí. V tomto prostředí lze využívat blokově orientované programování, které je podobné prostředí Scratch, ale lze v něm přepnout i do jiných programovacích jazyků, jako jsou Python, nebo JavaScript. Mezi těmito programovacími jazyky lze libovolně přecházet (přepínat si je), je možno je tedy i kombinovat. Pokud je to technicky proveditelné, vždy se dané úseky zobrazí ve vybraném programovacím jazyce. ([5]; str. 5)

Mezi další možnosti volby vývojového prostředí patří: <https://codewith.mu/>, které připomíná více klasické programovací prostředí. [14]



Obrázek 37: Ukázka části kódu v blokově orientovaném programování

```

1 input.onButtonPressed(Button.A, function () {
2   led.plot(0, 0)
3   led.plot(0, 2)
4   led.plotBrightness(2, 2, 150)
5 })
6 input.onButtonPressed(Button.B, function () {
7   led.plot(4, 2)
8   led.plot(4, 4)
9 })

```

Obrázek 38: Ukázka části kódu zapsaného v JavaScript

```

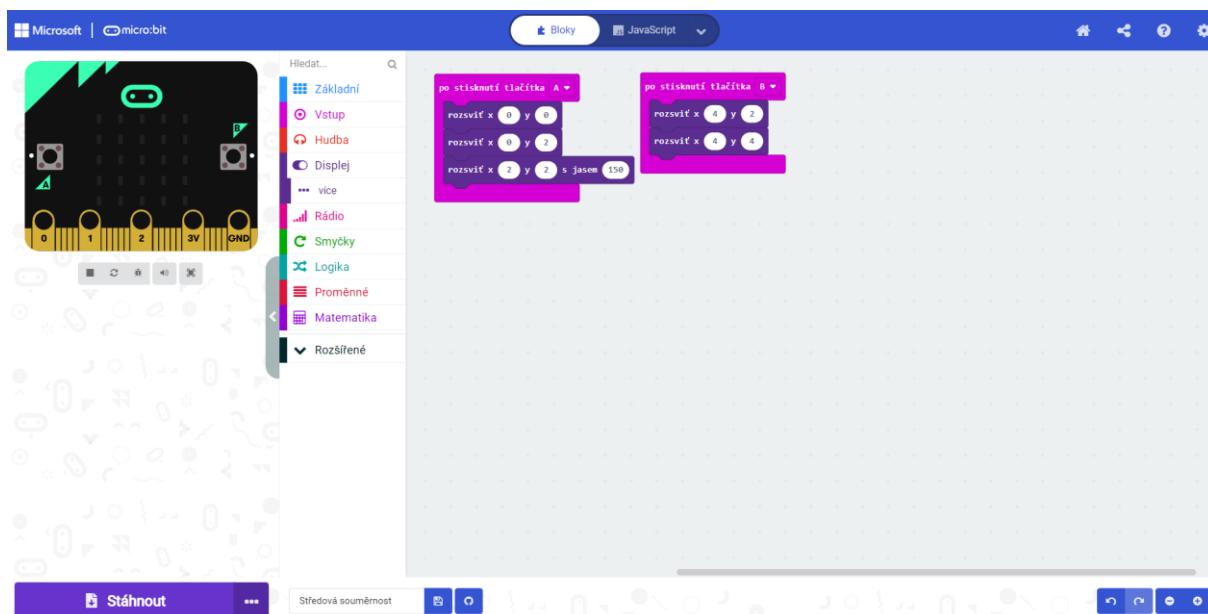
1 def on_button_pressed_a():
2   led.plot(0, 0)
3   led.plot(0, 2)
4   led.plot_brightness(2, 2, 150)
5 input.on_button_pressed(Button.A, on_button_pressed_a)
6
7 def on_button_pressed_b():
8   led.plot(4, 2)
9   led.plot(4, 4)
10 input.on_button_pressed(Button.B, on_button_pressed_b)
11

```

Obrázek 39: Ukázka části kódu zapsaného v Pythonu

Popis vývojového prostředí

Na obr. 40 je znázorněna ukázka webového vývojového prostředí <https://makecode.microbit.org/>. Toto vývojové prostředí vytvořila a volně zprostředkovává společnost Microsoft.



Obrázek 40: Vývojové prostředí www.makecode.microbit.org

Velkou výhodou tohoto vývojového prostředí jsou barevně odlišené příkazy, a to i po přepnutí z jednoho programovacího jazyka do jiného. Psaný text už však barevné odlišení nemá, barevně odlišené jsou pouze bloky.

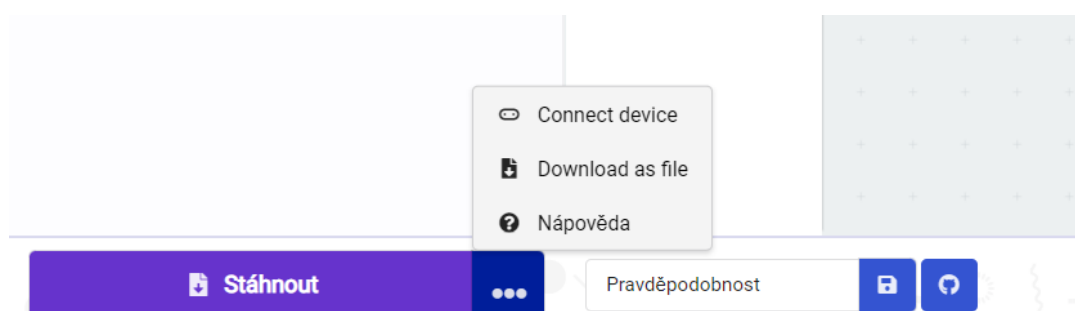
Vývojové prostředí je pro každý z jazyků více méně stejné, jediným rozdílem mezi blokově orientovaným prostředím a prostředím pro zápis kódu v Pythonu nebo JavaScriptu je plocha pro zápis kódů. V blokově orientovaném prostředí se na ploše spojují bloky, které mohou být umístěny libovolně, na rozdíl od toho v Pythonu nebo JavaScriptu jsou užity řádky pro zápis programu. Viz obrázky č. 37 – 39.

Vývojové prostředí se skládá z pěti částí:

1. Záhloví – V záhlaví nalezneme název micro:bit a také název společnosti Microsoft, která provozuje vývojové prostředí makecode.microbit.org, dále pak přepínání mezi programovacími jazyky a také nastavení vývojového prostředí, kde lze změnit například jazyk vývojového prostředí.
2. Simulační okno – V simulačním okně můžeme vidět zobrazený micro:bit. Toto simulační okno slouží ke zkontrolování funkčnosti sestaveného programu bez toho, abychom museli propojovat micro:bit s počítačem. Lze tedy programovat micro:bit i bez přítomnosti robota.
3. Knihovna bloků / příkazů – Knihovna je rozdělena do daného počtu složek, aby hledání bloků / příkazů bylo snadnější a aby bylo prostředí přehlednější.
4. Plocha pro zápis kódu

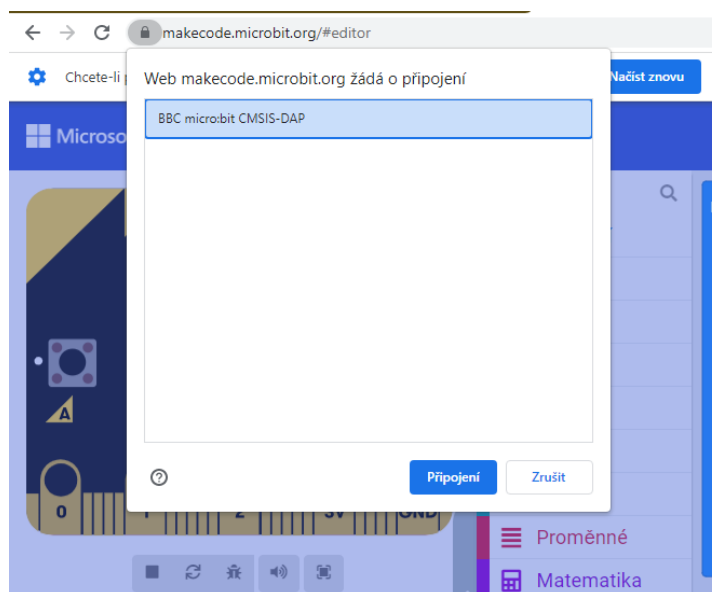
5. Zápatí – V zápatí se nachází možnost ukládání projektů (programů) do počítače, ale také možnost propojení micro:bitu s počítačem.

Propojení micro:bitu s počítačem probíhá v několika fázích, nejprve je potřeba jej propojit mikro USB kabelem s počítačem, poté v zápatí vývojového prostředí klikneme na tři tečky, z nabídky v zobrazeném plovoucím okně vybereme možnost „Connect device“.



Obrázek 41: Propojení micro:bitu s počítačem I

Následně zvolíme možnost „Dále“ ve dvou vyskakujících oknech a v poslední fázi vybereme zařízení micro:bit a klikneme na tlačítko „Připojení“.



Obrázek 42: Propojení micro:bitu s počítačem II

V zápatí se také nachází možnosti „Zpět“ nebo „Vpřed“, první z nich umožňuje vrátit nebo obnovit několik dříve proběhlých úkonů ve vývojovém prostředí. Jsou zde i tlačítka se znaménky „+“ a „–“, která zvětšují a zmenšují velikosti ve vývojovém prostředí (zvětšují / zmenšují velikost bloků, nebo písma).

3.3 Technické požadavky pro práci s micro:bity a pracovními listy

Pro práci s micro:bity a pracovními listy jsou potřeba v případě:

- distanční formy výuky
 - počítač s libovolným operačním systémem a připojením k internetu
- prezenční formy výuky
 - micro:bit
 - USB kabel s mikro USB zakončením
 - počítač s libovolným operačním systémem a připojením k internetu
 - pracovní listy

4 Mezipředmětové vztahy

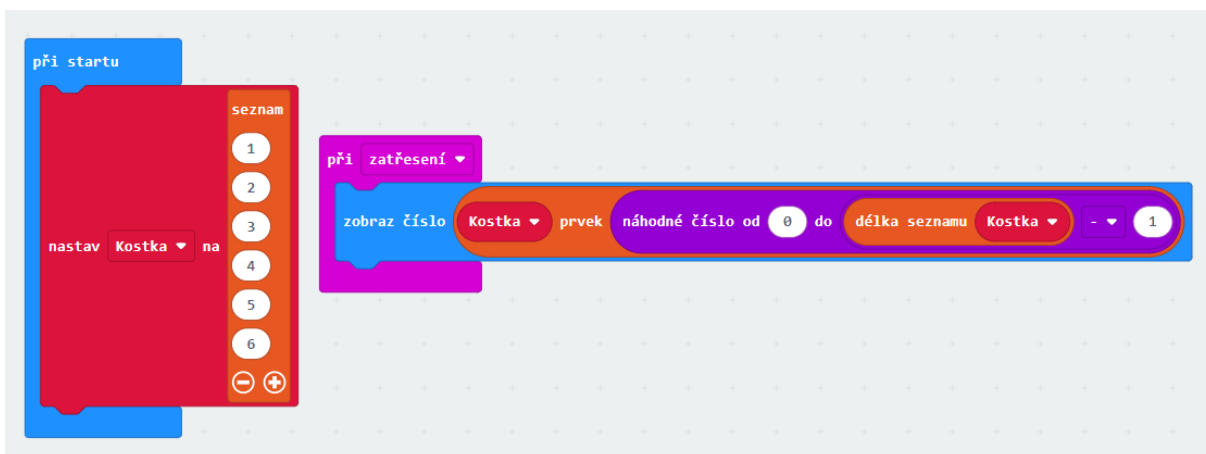
V současné době dochází k revizi Rámcového vzdělávacího programu pro základní vzdělávání. Jedním z cílů revize je mimo jiné zařazení více programování do hodin informatiky. Na druhou stranu je plánována redukce učiva v jiných vyučovaných předmětech. Z tohoto důvodu se jeví jako výhodné propojovat výuku školních předmětů všude tam, kde je to jen možné.

4.1 Využití micro:bitu v matematice

V rámci této diplomové práce se propojují poznatky hned ze dvou školních předmětů, a to z informatiky a z matematiky. Hlavní myšlenka propojení těchto dvou předmětů je v převádění řešení problému z jedné oblasti do druhé. Pro toto propojení byly vytvořeny speciální pracovní listy. Žáci v nich mají za úkol vyřešit matematické problémy pomocí naprogramování robota.

V každém pracovním listě je zadáný matematický problém, přesněji geometrický problém, jehož zadání musí žáci nejprve správně pochopit z geometrického hlediska a záhy pomocí svých znalostí z geometrie jej mají za úkol vyřešit. Poté musí problém rozdělit na dílčí části a následně musí vymyslet, jak jej předají k vyřešení robotovi.

Pomocí programování micro:bitu lze propojovat mnohem více oblastí z matematiky s informatikou. Naprogramováním jednoduchého programu, který náhodně vybírá číslo z pole, nebo vytvořením simulace hodu kostkou se dotýkáme oblasti pravděpodobnosti a kombinatoriky. Také by bylo možné využít micro:bit ve statistice, kde by žáci mohli získávat informace/data díky vestavěným funkcím micro:bitu, např. by mohli měřit hodnoty teplot, které by si zaznamenávali, a poté by data zpracovávali pomocí statistických metod (např. výpočet průměrné hodnoty, maximální hodnoty, minimální hodnoty, určení modusu, mediánu aj.).



Obrázek 43: Vytvořený kód pro simulaci hodu kostkou pomocí micro:bitu

Mimo jiné by díky možnostem využití logických relací bylo možné propojit informatiku s výrokovou logikou. Pomocí programování micro:bitu by žáci mohli vyhodnocovat a následně ověřovat pravdivostní hodnoty výroků, případně i složených výroků. Díky těmto úkonům by tedy žáci zjišťovali, které výroky sestavený program vyhodnocuje jako „false“ a které jako „true“.

4.2 Využití micro:bitu v dalších školních předmětech

Micro:bit lze využít v mnohem více školních předmětech, čímž je možné propojit informatiku kromě matematiky s dalšími předměty a vytvořit tak další různé mezipředmětové vztahy.

Micro:bit ve své základní výbavě umí měřit teplotu, a proto jej můžeme využít buď ve fyzice, nebo v chemii, když je potřeba zjišťovat teplotu různých předmětů či látek. Také je v něm zabudovaný kompas, který lze zakomponovat do hodin zeměpisu nebo do hodin tělesné výchovy, případně jej lze využít při sportovně-turistických kurzech.

Micro:bitem lze naprogramovat také např. stopky, které je opět možné využít ve fyzice, a to při fyzikálních pokusech (např. při měření rychlosti pohybujících se předmětů či těles), nebo během tělesné výchovy při měření např. rychlosti běhu žáků.

Kromě všech výše uvedených využití dokáže micro:bit také přehrávat tóny, a tak jej lze s výhodou zakomponovat do hodin hudební výchovy.

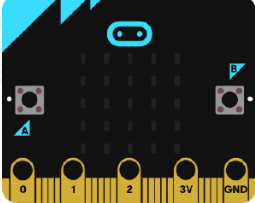
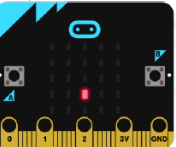
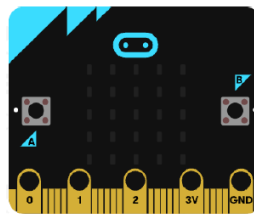
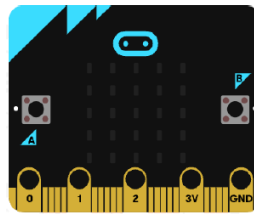
5 Pracovní listy a dotazník

5.1 Tvorba pracovních listů

V rámci této diplomové práce byly vytvořeny speciální pracovní listy k procvičování úloh z geometrie pomocí využití micro:bitu. Každý vytvořený pracovní list obsahuje minimálně dvě úlohy z jedné oblasti geometrie. Dříve, než začneme s žáky pracovat s pracovními listy obsahujícími úlohy k procvičování shodných zobrazení a analytické geometrie v rovině, vyplníme společně pracovní list s názvem „Úvodní hodina“. Ten žáky seznamuje s micro:bitem a vývojovým prostředím, ve kterém budou programovat.

Pracovní list určený pro úvodní hodinu práce s micro:bitem je možné použít jak na základní, tak na střední škole. Poukazuje především na jinak orientovaný kartézský souřadnicový systém, než na jaký jsou žáci ve škole běžně zvyklí.

Jako doprovodná aktivita k tomuto prvnímu pracovnímu listu je zařazení vlastního zkoumání vývojového prostředí žákem. Jako doprovodné úlohy může žák dostat za úkol např. vysvětlit nějaký obrazec, který se bude postupně měnit. Vyplnění úloh z tohoto pracovního listu, ale i úspěšné splnění doplňujících úloh by právě mělo vést k prozkoumání vývojového prostředí žáky.

Micro:bit	
https://makecode.microbit.org/	
1. úloha	
Jak funguje číslování řádků a sloupců?	
2. úloha	Vzorové řešení
Rozsviť bod na souřadnicích [1, 3]	Rozsviť bod na souřadnicích [2, 3]
Označ si, která dioda se rozsvítí?	Jak vypadá kód:
	
	Zházornění, kde se bod v matici LED diod zobrazí:
	
Existuje více možností, jak rozsvítit bod. Dokážeš najít nějaký další způsob?	
3. úloha	
Rozsviť bod na souřadnicích [2, 3] s jasnou 150.	
Označ si, která dioda se rozsvítí?	
	
4. úloha	
Po stisknutí tlačítka A rozsviť obvod žhvierce se stranou 3 jednotky* libovolně umístěný na Micro:bit	
* jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami	
Označ si, která dioda se rozsvítí?	
	

Obrázek 44: Pracovní list – Úvodní hodina

5.1.1 Pracovní listy pro žáky základní školy

Speciálně vytvořené pracovní listy pro žáky základní školy mají sloužit k úvodu do problematiky programování robotů, tj. k úvodu robotiky, či pouze k úvodu programování a zároveň jako opakování, nebo upřesnění konkrétních vybraných geometrických témat, přesněji shodných zobrazení v rovině.

Mezi další kritéria volby úloh zařazených do speciálně vytvořených pracovních listů byla jejich možná programovatelnost a také názornost při zobrazení na LED diodách micro:bitu.

Každý pracovní list vytvořený pro žáky základní školy obsahuje svůj název, slovní zadání úlohy, vysvícenými LED diodami znázorněné zadání úlohy na obrázku micro:bitu, vyobrazení bodů a geometrických objektů včetně jejich popisů na příslušných LED diodách micro:bitu, převedení problému na geometrický (tj. grafické zadání geometrické úlohy v rovině) a nakonec úlohy z programování pro sestavení odpovídajícího programu, který řeší danou úlohu.

Micro:bit - Osová souměrnost

1. úloha

Kliknutím na tlačítko A vybereš zadání pro zobrazení bodu H v dané osově souměrnosti. Najdi a rozsviť bod H' , který je osově souměrný s bodem H . Osa o souměrnosti je vyznačena slaběji.

Micro:bit

Geometrické objekty v micro:bitu

Převedení problému na geometrický

Napište souřadnice

Doplň souřadnice do kódu tak, aby se po kliknutí na tlačítko B zobrazil obraz H' bodu H .

rozsviť x y

Obrázek 45: Ukázka struktury pracovního listu určeného pro žáky základní školy

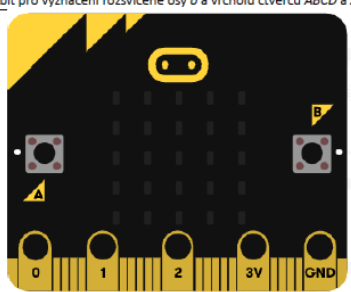
5.1.2 Pracovní listy pro žáky střední školy

Speciálně vytvořené pracovní listy pro žáky střední školy mají sloužit jako opakování vybraných oblastí geometrie a jako podpůrné prostředky pro výuku algoritmizace nebo programování.

Oproti pracovním listům vytvořeným pro žáky základní školy jsou pracovní listy určené pro žáky střední školy více zaměřené na programování.

V rámci zopakování jazyka JavaScript a zvyknutí si na vývojové prostředí makecode.microbit.org se mimo jiné využil pracovní list s názvem „Posunutí“, který je stejný i pro žáky základní školy.

Pracovní listy vytvořené pro žáky střední školy mají jednotnou strukturu, která spočívá v rozdělení oblasti pracovních listů na čtyři části: název pracovního listu, slovní zadání úlohy, obrázek micro:bitu, na němž si žáci mohou vyznačit zadané objekty, případně i hledané řešení a otázky s volným místem pro zápis příslušného kódu.

Micro:bit - Osová souměrnost
1. úloha
Vhodně umístí (rozsvítí) svislou osu o tak, aby bylo možné zobrazit v osově souměrnosti dané osou o libovolně umístěný čtverec ABCD se stranou délky 1 jednotka* na čtverec A'B'C'D'. Přitom čtverec ABCD umístíte tak, aby neměl žádný bod společný s osou o a aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy čtverce A'B'C'D'. Osu o rozsvítí s nižším jasnem než vrcholy obou čtverců. * jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru
Micro:bit pro vyznačení rozsvícené osy o a vrcholů čtverců ABCD a A'B'C'D'

Kód v JavaScript
Jak vypadá kód v JavaScript? Zapiš si, jak vypadají jednotlivé příkazy:

Obrázek 46: Ukázka pracovního listu určeného pro žáky střední školy

5.2 Vzorová řešení úloh z pracovních listů

V této části jsou podrobně ukázána a rozebrána vzorová řešení jednotlivých úloh zařazených v pracovních listech. Pokud má úloha více možností řešení, je ukázáno jedno z možných řešení a dále je uvedena poznámka o možné existenci dalšího řešení, případně dalších řešení.

5.2.1 Vzorová řešení úloh z pracovních listů určených pro žáky ZŠ

Všechny pracovní listy lze lehce modifikovat, a to především v případě, že by programování bylo pro žáky velmi náročné, nebo když by se dané úlohy daly využít v úvodu do programování za účelem vytvoření programu bez funkcí, i bez podmínek a proměnných.

Pomocí sestavených úloh si žáci mohou opakovat a procvičovat příklady zaměřené na shodná zobrazení v rovině a učit se postupně proměnným, podmínkám, cyklům a funkcím v programování. Je možné nechat žáky přijít s nějakým řešením a poté se dotazovat, jak program zapsat přehledněji, co by v tomto směru mohlo pomoci apod.

Osová souměrnost

1. úloha

Zadání: Kliknutím na tlačítko **A** vybereš zadání pro zobrazení bodu H v dané osově souměrnosti. Najdi a rozsviť bod H' , který je osově souměrný s bodem H . Osa o souměrnosti je vyznačena slaběji.

Převedení problému na geometrický:

Napište, na jakých souřadnicích se rozsvítí přenesený bod?

na souřadnicích 3,3

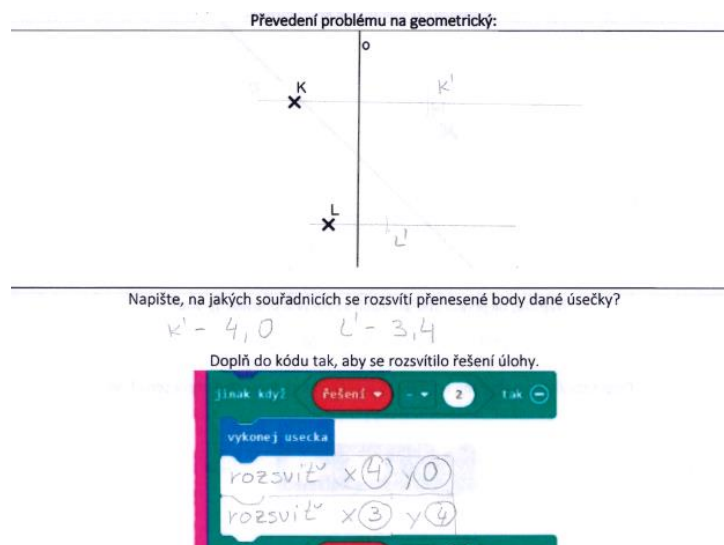
Doplň souřadnice do kódu tak, aby se po kliknutí na B zobrazil přenesený bod.

rozsviť x 3 y 3

Obrázek 47: Ukázka řešení 1. úlohy v pracovním listě Osová souměrnost

2. úloha

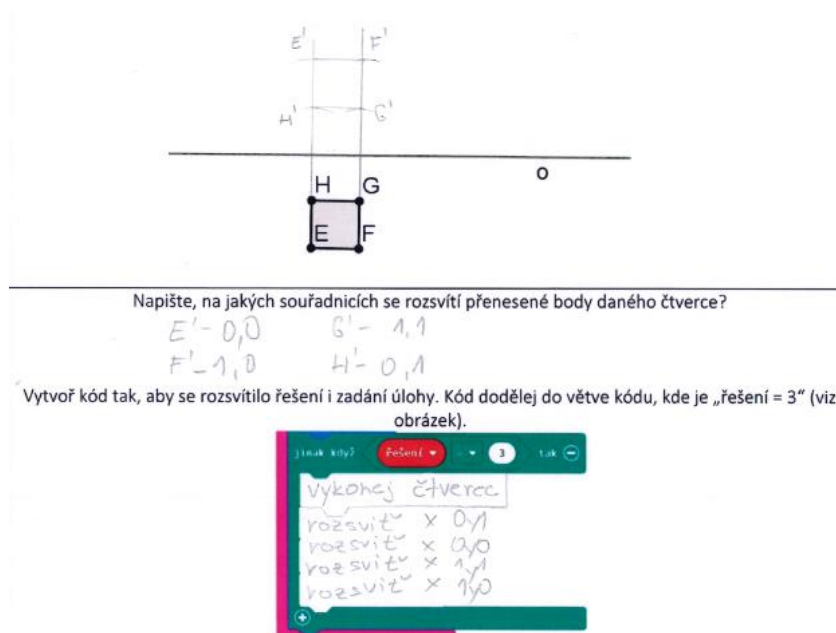
Zadání: Po druhém kliknutí na tlačítko **A** vybereš zadání pro zobrazení úsečky KL v dané osově souměrnosti. Zobraz úsečku KL v osově souměrnosti. Osa je vyznačena slaběji.



Obrázek 48: Ukázka řešení 2. úlohy v pracovním listě *Osová souměrnost*

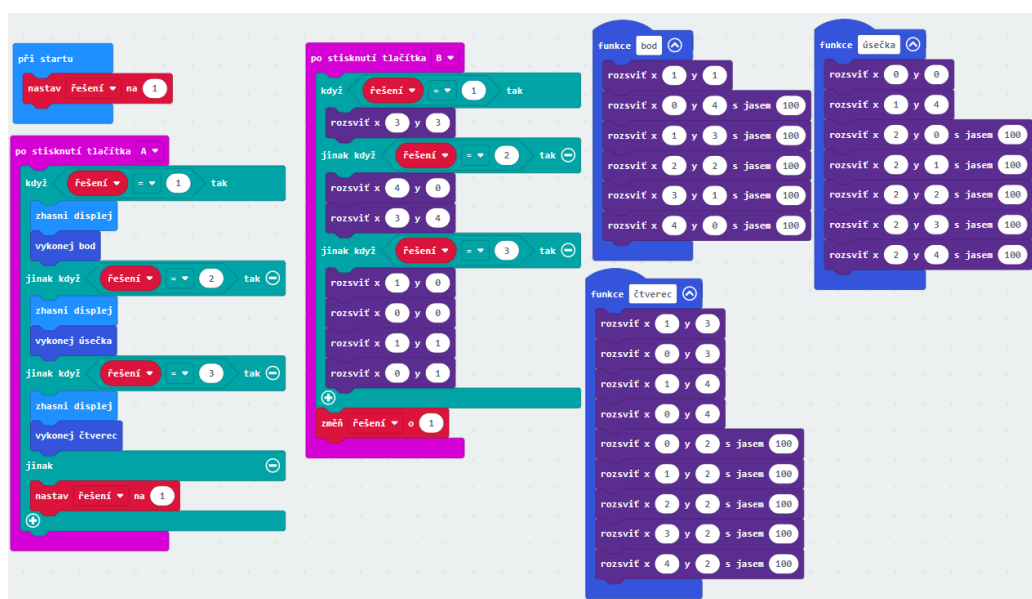
3. úloha

Zadání: Po třetím kliknutí na tlačítko **A** vybereš zadání pro zobrazení čtverce $EFGH$ v dané osově souměrnosti. Zobraz čtverec $EFGH$ v osově souměrnosti. Osa je vyznačena slaběji.



Obrázek 49: Ukázka řešení 3. úlohy v pracovním listě *Osová souměrnost*

Výsledný program:



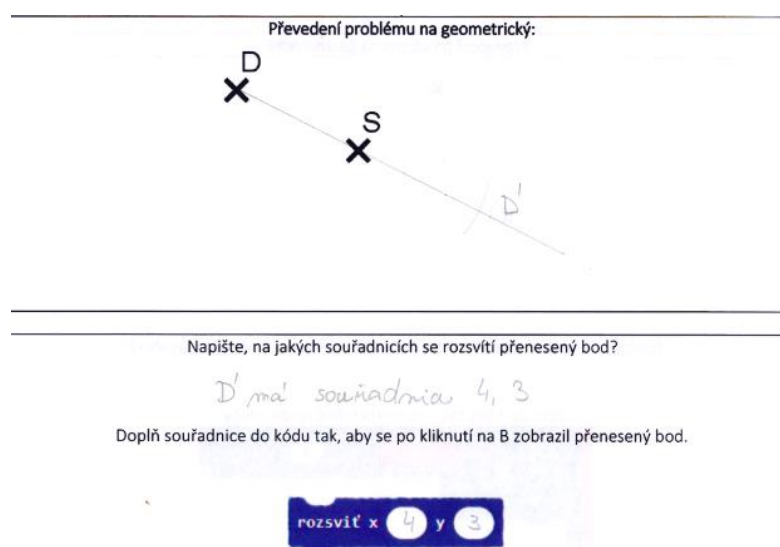
Obrázek 50: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu v názvem *Osová souměrnost*

Obrázek č. 50 ukazuje výsledný program, který využívá proměnných, podmínek a funkcí.

Středová souměrnost

1. úloha

Zadání: Kliknutím na tlačítko A vybereš zadání pro zobrazení bodu pomocí středové souměrnosti. Najdi bod D' , který je středově souměrný s bodem D . Střed S je vyznačen slaběji.

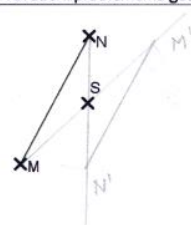


Obrázek 51: Ukázka řešení 1. úlohy v pracovním listě *Středová souměrnost*

2. úloha

Zadání: Po druhém kliknutí na tlačítko **A** vybereš zadání pro zobrazení úsečky MN v dané středové souměrnosti. Zobraz úsečku MN ve středové souměrnosti. Střed S je vyznačen slaběji.

Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí přenesené body dané úsečky?

$M' = 4,0$ $N' = 2,4$

Doplň do kódu tak, aby se rozsvítlo řešení úlohy.

jinak když? řešení 2 tak

vykonej usecka

rozsvit x 4 y 0

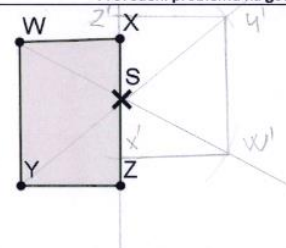
rozsvit x 2 y 4

Obrázek 52: Ukázka řešení 2. úlohy v pracovním listě Středová souměrnost

3. úloha

Zadání: Po třetím kliknutí na tlačítko **A** vybereš zadání pro zobrazení obdélníka $WYZX$ v dané středové souměrnosti. Zobraz obdélník $WYZX$ ve středové souměrnosti. Střed S je vyznačen slaběji.

Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí přenesené body daného obdélníka?

$x' = 2,3$ $z' = 2,0$
 $y' = 4,0$ $w' = 4,3$

Vytvoř kód tak, aby se rozsvítlo řešení i zadání úlohy. Kód dodělej do větve kódu, kde je „řešení = 3“ (viz obrázek).

jinak když? řešení 3 tak

rozsvit x 2 y 3

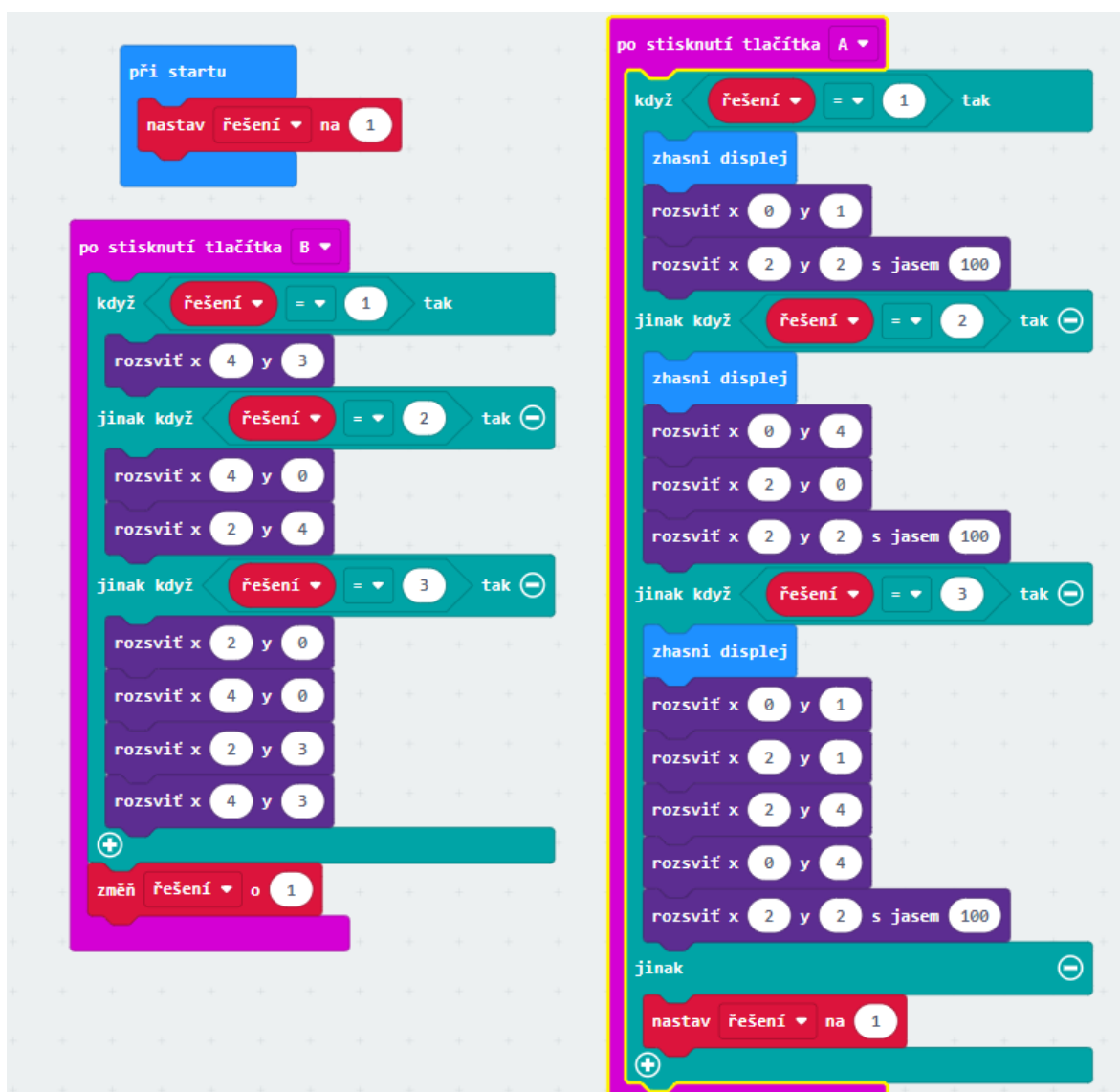
rozsvit x 2 y 0

rozsvit x 4 y 0

rozsvit x 4 y 3

Obrázek 53: Ukázka řešení 3. úlohy v pracovním listě Středová souměrnost

Výsledný program:



Obrázek 54: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem Středová souměrnost

V uvedeném řešení nebyly využity funkce, žáky je možné doprovodnými otázkami navést k vytvoření přehlednějšího programu, který využívá funkce.

Zobrazené řešení ukazuje využití podmínek a proměnných.

Posunutí

1. úloha

Zadání: Zobraz vodorovnou úsečku *OP*. Pro stisknutí tlačítka **A** vytvoř program, který bude každé 3 vteřiny posouvat úsečku *OP* vždy o jednu jednotku¹ ve svislém směru dolů.

¹ jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami ve vodorovném či ve svislém směru

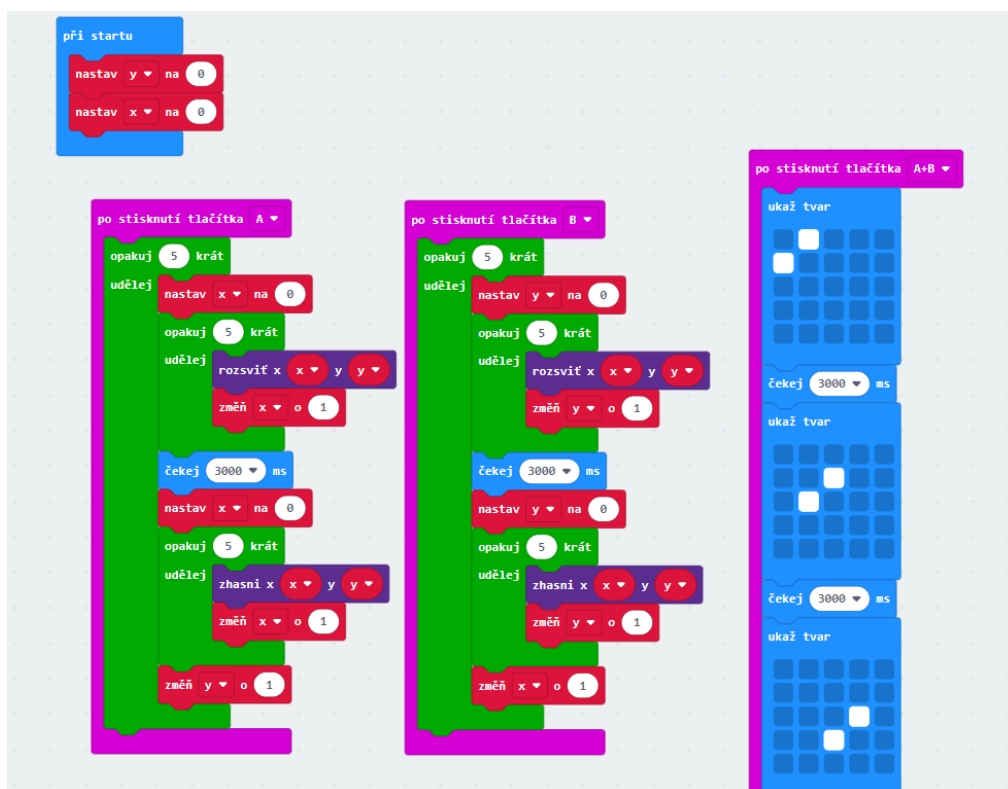
2. úloha

Zadání: Zobraz svislou úsečku UV . Pro stisknutí tlačítka **B** vytvoř program, který bude každé 3 vteřiny posouvat úsečku UV vždy o jednu jednotku¹ ve vodorovném směru doprava.

3. úloha

Zadání: Zobraz šikmou úsečku KL . Pro stisknutí obou tlačítek **A**, **B** vytvoř program, který bude každé 3 vteřiny posouvat úsečku KL vždy o jednu jednotku² v diagonálním (úhlopříčném) směru doprava dolů.

Výsledný program:



Obrázek 55: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem Posunutí

Pracovní list s názvem Posunutí by měl být zařazen jako poslední pracovní list. Při sestavování programu zobrazujícího řešení úlohy se totiž využívá vnořených cyklů, které mohou být pro žáky složité.

Proces posouvání úsečky skončí ve chvíli, kdy by se měly vysvítit LED diody mimo matici. Žákům lze úlohu rozšířit tím, že pokud by se měly rozsvítit LED diody mimo matici, vysvítí se opět první řádek.

² jednou jednotkou v diagonálním směru rozumíme složené posunutí o jednu jednotku ve vodorovném směru a o jednu jednotku ve svislém směru

Otočení

1. úloha

Zadání: Kliknutím na tlačítko **A** vybereš zadání pro otočení bodu. Otoč tento bod o $+90^\circ$. Střed otočení S je rozsvícen slaběji.

Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí otočený bod?

K' má souřadnice 0,4

Doplň souřadnice do kódu tak, aby se po kliknutí na B zobrazil otočený bod.

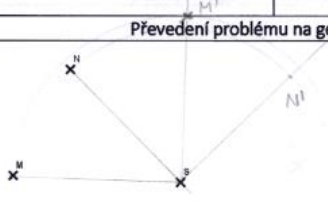
rozsvít' x 0 y 4

Obrázek 56: Ukázka řešení 1. úlohy v pracovním listě Otočení

2. úloha

Zadání: Po druhém kliknutí na tlačítko **A** se aktivuje zadání pro otočení úsečky MN . Otoč tuto úsečku o -90° . Střed otočení S je rozsvícen slaběji.

Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí otočené body dané úsečky?

Doplň do kódu tak, aby se rozsvítilo řešení úlohy.

jinak kdy? řešení - 2 tak

vykonej usecka

rozsvít' x 0 y 0

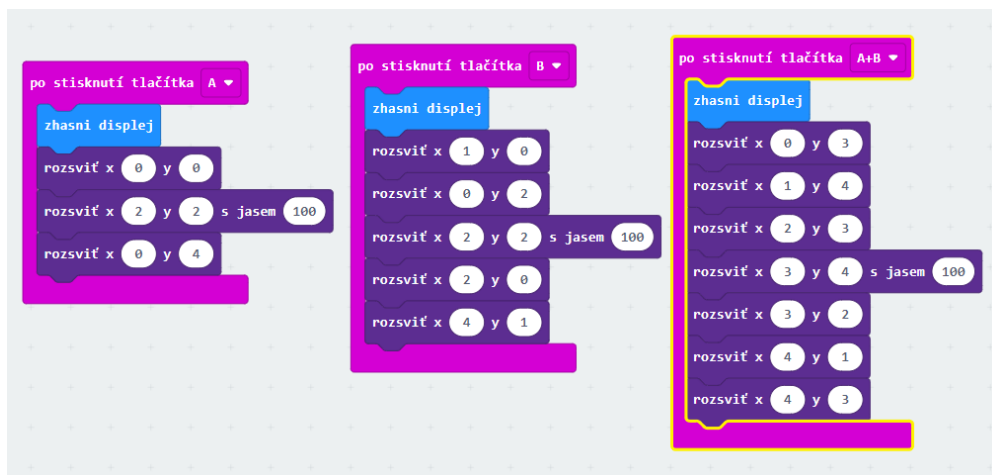
rozsvít' x 0 y 0

Obrázek 57: Ukázka řešení 2. úlohy v pracovním listě Otočení

3. úloha

Zadání: Po třetím kliknutí na tlačítko **A** se zobrazí zadání pro otočení trojúhelníku *OPQ*. Otoč tento trojúhelník o $+90^\circ$. Střed otočení *S* je rozsvícen slaběji.

Výsledný program:



Obrázek 58: Výsledný program zobrazující zadání a také řešení úloh z pracovního listu s názvem *Otočení*

Sestavený výsledný program, který zobrazuje v matici LED diod micro:bitu správná řešení jednotlivých úloh, viz obr. 58, neobsahuje ani proměnné, ani podmínky. Takovýto způsob zápisu programu je možné využít v případě, kdy žákům činí programování velké potíže. Tehdy se pomocí společné diskuze snažíme nalézt řešení, které je požadováno v pracovním listě. Tedy postupným klikáním na tlačítko **A** zobrazujeme zadání jednotlivých úloh a následným postupným stlačováním tlačítka **B** řešení příslušných úloh.

5.2.2 Vzorová řešení úloh z pracovních listů určených pro žáky SŠ

Speciálně vytvořené pracovní listy pro žáky střední školy lze také modifikovat. Když žáci jednotlivá zadání úloh v pracovních listech vyřeší, je možné jim přidat další rozšiřující problémy, např. takové, aby upravili kód tak, že skutečně nesmí záviset na umístění objektu, vzor a obraz musí být pevně svázaný nějakým konkrétním předpisem apod.

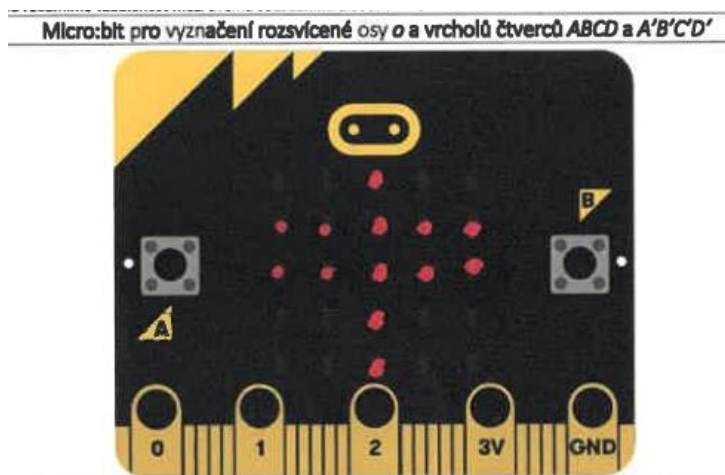
Kromě pracovního listu s názvem „Úvodní hodina“ by se jako první měl použít pracovní list určený i pro žáky základní školy s názvem „Posunutí“.

Ukázky vzorových řešení jednotlivých úloh jsou napsány v jazyce JavaScript.

Osová souměrnost

1. úloha:

Zadání: Vhodně umístí (rozsvítí) svislou osu o tak, aby bylo možné zobrazit v osově souměrnosti dané osou o libovolně umístěný čtverec $ABCD$ se stranou délky 1 jednotka³ na čtverec $A'B'C'D'$. Přitom čtverec $ABCD$ umístěte tak, aby neměl žádný bod společný s osou o a aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy čtverce $A'B'C'D'$. Osu o rozsviďte s nižším jasnem než vrcholy obou čtverců.



Obrázek 59: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Osová souměrnost

Výsledný kód:

```
/*Souřadnice jednoho z vrcholů čtverce s délkou strany 1
j; měnitelná*/
let ay = 0
let ax = 0
/*Souřadnice prvního bodu osy; měnitelná*/
let y = 0
let x = 2
/* tělo kódu */
for (let index = 0; index < 5; index++) {
    led.plotBrightness(2, y, 100)
    y += 1
}
led.plot(ax, ay)
led.plot(ax + 1, ay + 1)
led.plot(ax, ay + 1)
```

³ jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru

```

led.plot(ax + 1, ay)
led.plot(4 - ax, ay)
led.plot(4 - (ax + 1), ay + 1)
led.plot(4 - ax, ay + 1)
led.plot(4 - (ax + 1), ay)

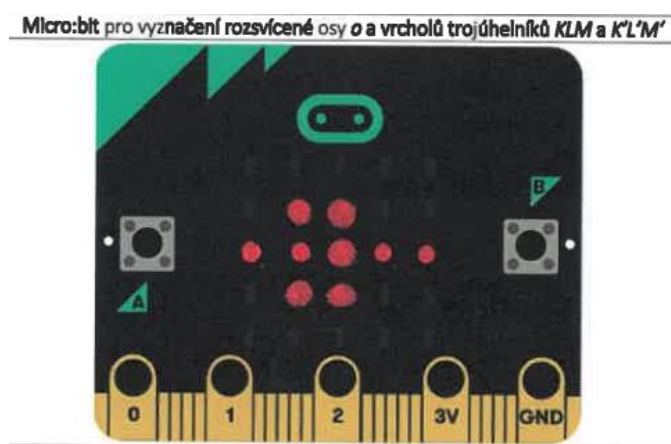
```

Úloha je vyřešena pomocí proměnných a cyklů tak, aby bylo možné zvolit libovolné umístění čtverce. Bylo by také možné doplnit podmínky, pokud by se zadala souřadnice vyšší než 4.

2. úloha

Zadání: Vhodně umístí (rozsvítí) osu o tak, aby se v osové souměrnosti dané osou o dal zobrazit libovolně umístěný obecný trojúhelník KLM , který má s osou o společný právě jeden bod. Přitom trojúhelník KLM umístí tak, aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy trojúhelníku $K'L'M'$, který je obrazem trojúhelníku KLM . Osu o rozsvítí s nižším jasnem než vrcholy obou trojúhelníků.

Vytvoř cyklus, který třikrát rozsvítí a zhasne trojúhelník KLM i jeho obraz $K'L'M'$.



Obrázek 60: Ukázka vyznačení zadání a řešení v micro:bitu - 2. úloha v pracovním listě Osová souměrnost

Výsledný kód:

```

// volitelné souřadnice jednoho vrcholu trojúhelníku
let kx = 1
let ky = 1
//souřadnice jednoho volitelného bodu osy
let x = 0
let y = 2
// tělo kódu

```

```

for (let index = 0; index < 5; index++) {
    led.plotBrightness(x, y, 100)
    x += 1
}
for (let index = 0; index < 3; index++) {
    led.plot(kx, ky)
    led.plot(kx, ky + 1)
    led.plot(kx + 1, ky)
    led.plot(kx, 4 - ky)
    led.plot(kx, 4 - (ky + 1))
    led.plot(kx + 1, 4 - ky)
    basic.pause(500)
    led.unplot(kx, ky)
    led.unplot(kx, ky + 1)
    led.unplot(kx + 1, ky)
    led.unplot(kx, 4 - ky)
    led.unplot(kx, 4 - (ky + 1))
    led.unplot(kx + 1, 4 - ky)
    led.plotBrightness(1, 2, 100)
    basic.pause(500)
}

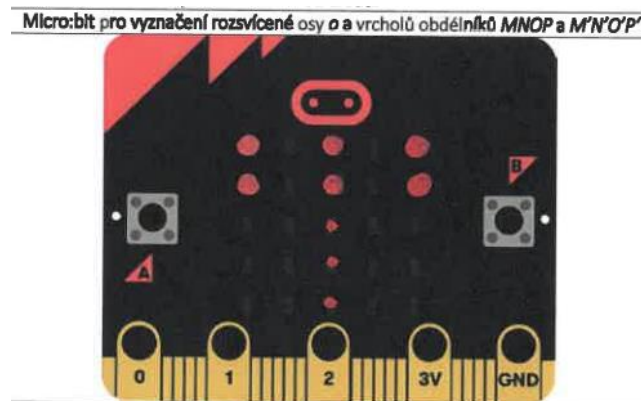
```

Úloha byla řešena tak, aby bylo možné v matici LED diod micro:bitu změnit umístění trojúhelníku a jeho obrazu v dané osově souměrnosti.

3. úloha

Zadání: Vhodně umísti (rozsvít') osu o tak, aby se v osově souměrnosti dané osou o dal zobrazit libovolně umístěný obdélník $MNOP$, který má s osou o společné právě dva vrcholy. Přitom obdélník $MNOP$ umísti tak, aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy jeho obrazu $M'N'O'P'$. Osu o rozsvít' s nižším jasnem než vrcholy obou obdélníků.

Postupně rozsvít' a zhasni diody na obvodu celého obdélníku tak, aby vždy svítla právě jedna dioda. Dále rozsvít' a zhasni osově souměrnou diodu k právě rozsvícené vzorové diodě.



Obrázek 61: Ukázka vyznačení zadání a řešení v micro:bitu - 3. úloha v pracovním listě Osová souměrnost

Výsledný kód:

```
//volitelná souřadnice jednoho vrcholu obdélníku a
volitelné velikosti stran obdélníku
let mx = 0
let my = 0
let m = 2
let n = 2
// první souřadnice volitelné osy
let x = 2
let y = 0
// tělo kódu
for (let index = 0; index < 5; index++) {
  led.plotBrightness(x, y, 100)
  y += 1
  basic.pause(100)
}
loops.everyInterval(500, function () {
  led.plot(mx, my)
  led.plot(4 - mx, my)
  basic.pause(500)
  led.unplot(mx, my)
  led.unplot(4 - mx, my)
  basic.pause(500)
  led.plot(mx + m, my)
  led.plot(4 - (mx + m), my)
  basic.pause(500)
```

```

    led.unplot(mx + m, my)
    led.unplot(4 - (mx + m), my)
    led.plotBrightness(mx + m, my, 100)
    basic.pause(500)
    led.plot(mx + m, my + n)
    led.plot(4 - (mx + m), my + n)
    basic.pause(500)
    led.unplot(mx + m, my + n)
    led.unplot(4 - (mx + m), my + n)
    led.plotBrightness(mx + m, my + n, 100)
    basic.pause(500)
    led.plot(mx, my + n)
    led.plot(4 - mx, my + n)
    basic.pause(500)
    led.unplot(mx, my + n)
    led.unplot(4 - mx, my + n)
    basic.pause(500)
  })

```

I tato úloha má opět volitelné vstupní parametry. Pro opakované rozsvěcení a zhasínání diod bylo využito opakování po určitém čase.

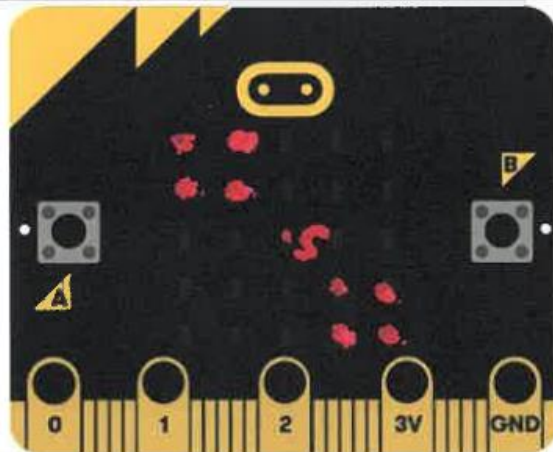
Středová souměrnost

1. úloha

Zadání: Vhodně umísti (rozsviť) střed souměrnosti S tak, aby se ve středové souměrnosti dané středem S dal zobrazit libovolně umístěný čtverec $ABCD$ se stranou délky 1 jednotka⁴ na čtverec $A'B'C'D'$. Přitom střed S zvol tak, aby nebyl bodem čtverce $ABCD$ a aby byly rozsvíceny všechny vrcholy čtverce $ABCD$ a jeho obrazu $A'B'C'D'$. Střed S rozsviť s nižším jasem než vrcholy obou čtverců.

Dále vytvoř kód, který po stisknutí tlačítka **A** změní velikost strany čtverce o jednu jednotku, a to postupně dvakrát, a po třetím stisknutí tlačítka **A** se opět zobrazí čtverec se stranou 1 jednotka.

⁴ jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru



Obrázek 62: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Středová souměrnost

Výsledný kód:

```
// souřadnice jednoho vrcholu čtverce a délka jeho
strany
let ay = 0
let ax = 0
let velikost = 0
velikost = 1
// volitelný střed středové souměrnosti
x = 2
y = 2
// tělo kódu
led.plotBrightness(x, y, 100)
led.plot(ax, ay)
led.plot(ax + 1, ay)
led.plot(ax, ay + 1)
led.plot(ax + 1, ay + 1)
led.plot(4 - ax, 4 - ay)
led.plot(4 - (ax + 1), 4 - ay)
led.plot(4 - ax, 4 - (ay + 1))
led.plot(4 - (ax + 1), 4 - (ay + 1))
// po stisknutí tlačítka A
input.onButtonPressed(Button.A, function () {
    basic.clearScreen()
    velikost += 1
})
```

```

        led.plotBrightness(x, y, 100)
        led.plot(ax, ay)
        led.plot(ax + velikost, ay)
        led.plot(ax, ay + velikost)
        led.plot(ax + velikost, ay + velikost)
        led.plot(4 - ax, 4 - ay)
        led.plot(4 - (ax + velikost), 4 - ay)
        led.plot(4 - ax, 4 - (ay + velikost))
        led.plot(4 - (ax + velikost), 4 - (ay + velikost))
    t))

    if (velikost == 3) {
        velikost = 0
    }

    })

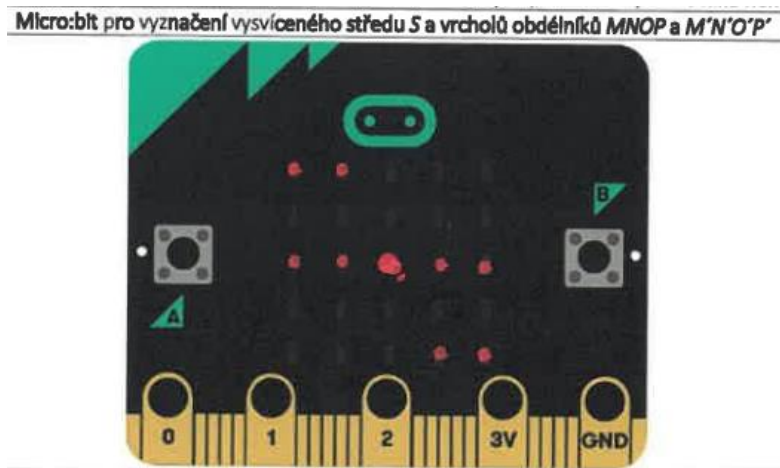
```

Úloha má opět volitelné parametry, jedním z volitelných parametrů jsou souřadnice jednoho z vrcholů daného čtverce, druhým volitelným parametrem je délka strany daného čtverce, třetím a posledním volitelným parametrem je umístění středu souměrnosti S v matici LED diod micro:bitu při dodržení podmínek zadání úlohy. Sestavený kód využívá proměnných a podmínek.

2. úloha

Zadání: Vhodně umísti (rozsviť) střed souměrnosti S tak, aby se ve středové souměrnosti dané středem S dal zobrazit libovolně umístěný obdélník $MNOP$. Přitom zvol střed S tak, aby nebyl bodem obdélníku $MNOP$ a aby byly rozsvíceny všechny vrcholy obdélníku $MNOP$ i jeho obrazu $M'N'O'P'$. Střed S rozsviť s nižším jasnem než vrcholy obou obdélníků.

Dále vytvoř kód, který bude postupně měnit pozici obdélníku tak, aby se rozměry obdélníku neměnily.



Obrázek 63: Ukázka vyznačení zadání a řešení v micro:bitu - 2. úloha v pracovním listě Středová souměrnost

Výsledný kód:

```
//souřadnice jednoho vrcholu obdélníku a délky jeho stran
let ay = 0
let ax = 0
let m = 1
let n = 2
// volitelný střed středové souměrnosti
let x = 2
let y = 2
// tělo kódu
led.plotBrightness(x, y, 100)
led.plot(ax, ay)
led.plot(ax + m, ay)
led.plot(ax, ay + n)
led.plot(ax + m, ay + n)
led.plot(4 - ax, 4 - ay)
led.plot(4 - (ax + m), 4 - ay)
led.plot(4 - ax, 4 - (ay + n))
led.plot(4 - (ax + m), 4 - (ay + n))
basic.pause(500)
for (let index = 0; index < 2; index++) {
  led.unplot(ax, ay)
  led.unplot(ax + m, ay)
  led.unplot(ax, ay + n)
```

```

led.unplot(ax + m, ay + n)
led.unplot(4 - ax, 4 - ay)
led.unplot(4 - (ax + m), 4 - ay)
led.unplot(4 - ax, 4 - (ay + n))
led.unplot(4 - (ax + m), 4 - (ay + n))
basic.pause(500)
ay += 1
led.plot(ax, ay)
led.plot(ax + m, ay)
led.plot(ax, ay + n)
led.plot(ax + m, ay + n)
led.plot(4 - ax, 4 - ay)
led.plot(4 - (ax + m), 4 - ay)
led.plot(4 - ax, 4 - (ay + n))
led.plot(4 - (ax + m), 4 - (ay + n))
basic.pause(500)
}

```

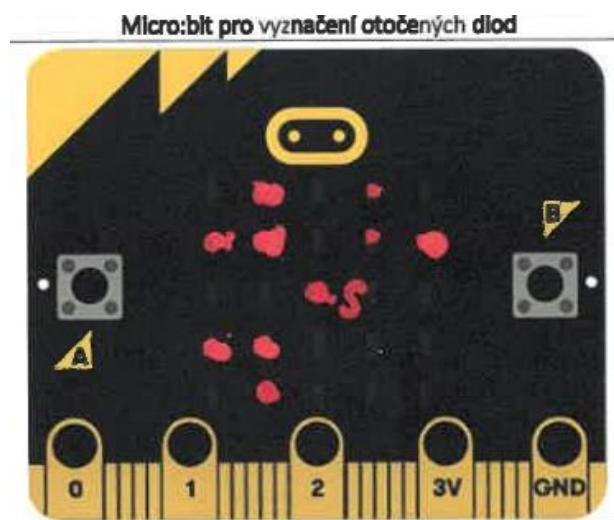
Úloha má volitelné vstupní parametry, jakými jsou rozměry obdélníku, souřadnice jednoho z vrcholů obdélníku a umístění středu souměrnosti S v matici LED diod micro:bitu.

V sestaveném programu řešení úlohy se aplikují znalosti proměnných a cyklu.

Otočení

1. úloha

Zadání: Umístí střed otočení S na souřadnice $[2, 2]$ micro:bitu. Střed S rozsvít s nižším jasem. Dále rozsvít vrcholy libovolného trojúhelníku ABC . Vytvoř kód tak, aby se kliknutím na tlačítko **A** otočil trojúhelník ABC kolem středu S o orientovaný úhel $+90^\circ$ a aby se kliknutím na tlačítko **B** trojúhelník ABC otočil o orientovaný úhel -90° . Přitom umístěte trojúhelník ABC na matici LED diod micro:bitu tak, aby všechny jeho vrcholy i vrcholy jeho obrazů byly rozsvíceny.



Obrázek 64: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Otočení

Výsledný kód:

```

zadání()
input.onButtonPressed(Button.A, function () {
    basic.clearScreen()
    zadání()
    led.plot(3, 0)
    led.plot(3, 1)
    led.plot(4, 1)
})
input.onButtonPressed(Button.B, function () {
    basic.clearScreen()
    zadání()
    led.plot(0, 3)
    led.plot(1, 3)
    led.plot(1, 4)
})
function zadání() {
    led.plotBrightness(2, 2, 100)
    led.plot(1, 0)
    led.plot(0, 1)
    led.plot(1, 1)
}

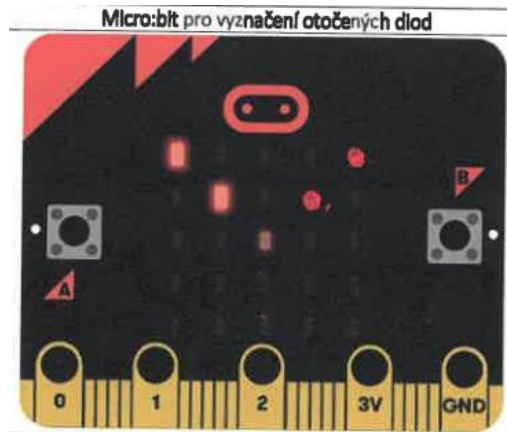
```

Kód vytvořený pro zobrazení řešení zadané úlohy ukazuje využití funkce.

2. úloha

Zadání: Umístí střed otočení S na souřadnice $[2, 2]$ micro:bitu. Střed S rozsvít s nižším jasem. Dále s vyšším jasem rozsvít diody na souřadnicích $[0, 0]$, $[1, 1]$ micro:bitu a vytvoř cyklus, který bude postupně rozsvěcet a zhasínat dvě dané diody otáčející se kolem středu S ve zvoleném směru (tj. buď po směru, nebo proti směru hodinových ručiček).

O kolik úhlových stupňů se v každém kroku každá z diod otočí? [Správná odpověď: Každá z diod se v každém kroku otočí o 90° .]



Obrázek 65: Ukázka vyznačení bodů v micro:bitu

Výsledný kód:

```
led.plotBrightness(2, 2, 100)
let x1 = 0
let y1 = 0
let x2 = 1
let y2 = 1
basic.forever(function () {
  if (x1 <= 4 && y1 == 0) {
    if (y1 < 0) {
      y1 = 0
      y2 = 1
    }
    led.plot(x1, y1)
    led.plot(x2, y2)
    basic.pause(500)
    led.unplot(x1, y1)
    led.unplot(x2, y2)
    x1 += 4
  }
```

```

        x2 += 2
    } else if (x1 >= 4 && y1 < 4) {
        if (x1 > 4) {
            x1 = 4
            x2 = 3
        }
        y1 += 4
        y2 += 2
        led.plot(x1, y1)
        led.plot(x2, y2)
        basic.pause(500)
        led.unplot(x1, y1)
        led.unplot(x2, y2)
    } else if (x1 > 0 && y1 >= 4) {
        if (y1 > 4) {
            y1 = 4
            y2 = 3
        }
        x1 += -4
        x2 += -2
        led.plot(x1, y1)
        led.plot(x2, y2)
        basic.pause(500)
        led.unplot(x1, y1)
        led.unplot(x2, y2)
    } else if (x1 == 0 && y1 > 0) {
        if (x1 < 0) {
            x1 = 0
            x2 = 1
        }
        y1 += -4
        y2 += -2
        led.plot(x1, y1)
        led.plot(x2, y2)
        basic.pause(500)
    }

```

```

        led.unplot(x1, y1)
        led.unplot(x2, y2)
    }
})

```

Vytvořený kód znázorňující postupně jednotlivé kroky řešení úlohy využívá cykly, podmínky a proměnné.

Analytická geometrie

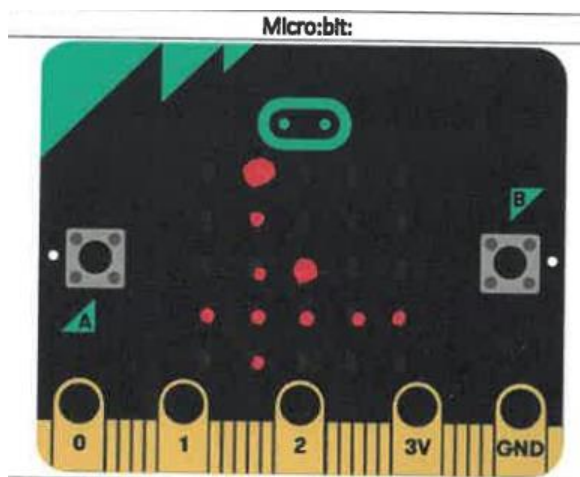
Úlohy jsou v této části zaměřené na práci se souřadnicovými systémy. Žák musí zadání i řešení úloh v této části uvažovat v kartézském souřadnicovém systému a zároveň musí obojí převádět do souřadnicového systému v micro:bitu.

Pracovní list by měl být zaměřen na opakování programování a dále by měl sloužit k obohacení výuky analytické geometrie, protože nabízí názornou vizuální ukázkou zobrazení zadaných a hledaných geometrických objektů díky rozsvěcujícím se diodám.

1. úloha

Zadání: Kliknutím na tlačítko **A** se zobrazí souřadnicový systém.

- Uprav kód tak, aby se osa x a osa y zobrazily s jasem 155.
- Rozsviť body K a L , kde $K = [2, 2]$ a $L = [1, -2]$
- Napiš obecnou rovnici přímky procházející body K a L .



Obrázek 66: Ukázka vyznačení zadání a řešení v micro:bitu - 1. úloha v pracovním listě Analytická geometrie

Výsledný upravený kód:

```

let x1 = 0
let y1 = 2
for (let index = 0; index < 5; index++) {
    led.plotBrightness(x1, y1, 155)
    x1 += 1
}
let x2 = 1
let y2 = 0
for (let index = 0; index < 5; index++) {
    led.plotBrightness(x2, y2, 155)
    y2 += 1
}
led.plot(3, 0)
led.plot(2, 4)

```

Obecná rovnice přímky:

Postup sestavení obecné rovnice přímky $p \equiv KL$ – určení obecné rovnice přímky p s užitím obecného zápisu obecné rovnice přímky a incidence bodů K, L a přímky p :

$$\frac{p \equiv KL: ax + by + c = 0}{K \in p: 2a + 2b + c = 0} \quad (1)$$

$$L \in p: a - 2b + c = 0 \quad (2)$$

Rovnici (2) odečteme od rovnice (1) a obdržíme:

$$a + 4b = 0 \quad (3)$$

Souřadnice směrového vektoru přímky p určíme pomocí rozdílu odpovídajících souřadnic jeho krajních bodů K, L , tj. $\mathbf{s} = K - L = (2 - 1, 2 - (-2)) = (1, 4)$. Ke směrovému vektoru přímky p určíme příslušný normálový vektor této přímky, a to tak, že zaměníme pořadí souřadnic směrového vektoru přímky a jednu z nich vynásobíme číslem (-1) . Normálový vektor přímky p má tedy souřadnice $\mathbf{n} = (4, -1)$ (nebo $(-4, 1)$).

Souřadnice normálového vektoru dosadíme do rovnice (3) a získáváme obecnou rovnici přímky p ve tvaru

$$4x - y + c = 0. \quad (4)$$

Do rovnice (4) dosadíme buď souřadnice bodu K (nebo bodu L), tj.

$$4 \cdot 2 - 2 + c = 0, \quad (5)$$

odkud

$$c = -6. \quad (6)$$

Obecná rovnice přímky p je tedy tvaru:

$$4x - y - 6 = 0. \quad (7)$$

2. úloha

Zadání: Umísti vhodně souřadnicový systém, který se zobrazí po kliknutí na tlačítko **B**.

a) Tento systém rozsvít s nižším jasnem než navzájem různé body Z a D , které libovolně umístí na přímku p .

b) Přímka p je dána parametrickým vyjádřením:

$$x = 2 - t$$

$$y = 1 + 2t, \quad t \in \mathbf{R}.$$

Přímku p po druhém stisknutí tlačítka **B** rozsvít.

Výsledný kód:

```
let y2 = 0
let x2 = 0
let y1 = 0
let x1 = 0
let kliknuti = 0
input.onButtonPressed(Button.B, function () {
  if (kliknuti == 0) {
    x1 = 0
    y1 = 3
    for (let index = 0; index < 5; index++) {
      led.plotBrightness(x1, y1, 155)
      x1 += 1
    }
    x2 = 1
    y2 = 0
  }
})
```

```

        for (let index = 0; index < 5; index++) {
            led.plotBrightness(x2, y2, 155)
            y2 += 1
        }
        kliknuti += 1
    } else if (kliknuti == 1) {
        led.plot(1, 0)
        led.plot(2, 2)
        led.plot(3, 4)
    }
})

```

5.3 Testování pracovních listů

5.3.1 Testování pracovních listů uskutečněné na ZŠ

Na základě ŠVP Základní školy Vrchlického v Liberci a po poradě s tamní paní učitelkou matematiky bylo rozhodnuto, že speciálně sestavené pracovní listy jsou vzhledem ke své obsahové náplni vhodné pro žáky 8. třídy. Testované oblasti z geometrie se totiž probírají až ve druhém pololetí 7. třídy.

Testování byli žáci tříd 8.A a 8.B ZŠ Vrchlického v Liberci. Nakonec nebyli testováni všichni žáci obou tříd, ale pouze jen vybraní žáci, přesněji jen ti šikovnější, u kterých byl po konzultaci s jejich vyučujícími alespoň nějaký předpoklad, že úlohy z pracovních listů zvládnou vypracovat.

Žáci obou testovaných tříd již dříve v hodinách informatiky absolvovali základy programování, a to v rámci studia blokově orientovaného jazyka Scratch.

Testování vytvořených pracovních listů probíhalo během hodin výpočetní techniky v 8. třídách.

Průběh testování pracovních listů

Testování pracovních listů probíhalo během 6 vyučovacích hodin ve třídě 8.B. Během první vyučovací hodiny se žáci seznámili s vývojovým prostředím makecode.microbit.org a se samotným micro:bitem. V průběhu první vyučovací hodiny žáci postupovali podle úloh zařazených v pracovním listě s pracovním názvem „Úvodní hodina“. Důležitou

součástí představení mikropočítače s názvem micro:bit bylo žákům ukázat, jak se micro:bit propojuje s počítačem.

Ve druhé vyučovací hodině žáci začali vypracovávat pracovní list s názvem „Osová souměrnost“. Zprvu bylo zapotřebí s žáky základní principy a vlastnosti osově souměrnosti zopakovat. Poté pro ně bylo nalezení správných řešení již snadné. Nejprve programovali každou úlohu separovaně tak, aby se řešení každé dílčí úlohy zobrazilo po stisknutí jiného tlačítka. Nakonec vymysleli algoritmus, který rozsvítí po určitém kliknutí řešení požadované úlohy. Vypracovávání úloh z pracovního listu s názvem „Osová souměrnost“ zasáhlo částečně i do 3. vyučovací hodiny, při níž žáci také již začali řešit a programovat úlohy z dalšího pracovního listu s názvem „Středová souměrnost“. Poté se v následujících vyučovacích hodinách postupně pokračovalo s vypracováváním dalších pracovních listů. Vždy po dokončení příslušného pracovního listu byl každý žák požádán o vyplnění dotazníku zjišťujícího zpětnou vazbu k příslušnému vypracovávanému pracovnímu listu. Na začátku vyplňování každého dalšího pracovního listu se stručně zopakovalo příslušné téma z geometrie.

Jako velmi zajímavý poznatek shledávám, že zobrazení geometrických řešení na ponechaných místech v pracovních listech využilo pouze několik málo žáků a někteří z nich dokonce jen u některých úloh. Důvodem může být také skutečnost, že žáci pracovní listy vyplňovali v hodinách informatiky, a ne v hodinách matematiky. Jinými slovy řečeno, žáci u sebe většinou neměli rýsovací pomůcky.

Ve třídě 8.A se stihla pouze úvodní hodina věnovaná představení vývojového prostředí `makecode.microbit.org` a micro:bitu. S žáky této třídy proběhlo pouze vypracování pracovního listu s názvem „Osová souměrnost“. Důvodem nepokračování v testování dalších pracovních listů v této třídě bylo časté odpadání hodin informatiky v této třídě.

5.3.2 Testování pracovních listů uskutečněné na SŠ

Na střední škole probíhalo testování pracovních listů ve třídách Střední průmyslové školy strojní a elektrotechnické a Vyšší odborné školy v Liberci (dále jen SPŠSE a VOŠ v Liberci), které byly svým učebním programem zaměřené na informační technologie a jejichž žáci měli k programovacím jazykům velmi blízko. Vzhledem k tomu, že se na příslušné SŠ nevyučoval programovací jazyk Python, žáci programovali a také vypracovávali úlohy v pracovních listech v jazyce JavaScript.

Testování pracovních listů vytvořených pro žáky SŠ probíhalo ve třetím ročníku SPŠSE a VOŠ v Liberci oboru „Informační technologie“ v rámci předmětu „Programování“ a poté i v prvním ročníku téže SPŠSE a VOŠ v Liberci také stejného studijního oboru „Informační technologie“, tentokrát však v rámci předmětu „Základy algoritmizace“.

S panem učitelem, který je vyučujícím obou výše zmíněných předmětů, jsme se shodli, že úlohy zařazené do pracovních listů obsahují problémy, které jsou velmi vhodné pro úvod do programování, jelikož tyto problémy obsahují pro žáky známá témata. Zároveň lze problémy rozložit na více podproblémů, které je třeba pro nalezení správného výsledku postupně vyřešit.

Pracovní listy jsem ve třetím ročníku SPŠSE a VOŠ v Liberci zkoušela s žáky osobně v rámci svého studijního předmětu „Souvislá praxe“ a v prvním ročníku SPŠSE a VOŠ v Liberci zadával žákům úlohy v pracovních listech sám pan učitel.

Jediným netestovaným pracovním listem je pracovní list s úlohami z analytické geometrie, protože v období testování pracovních listů nedošlo k průniku mezi obsahy výuky informatiky a matematiky. Jinak řečeno, do doby výzkumu a ani během výzkumu se v žádné ze dvou tříd, ve kterých testování probíhalo, dosud nevyučovala analytická geometrie.

Průběh testování pracovních listů

Osobně jsem pracovní listy zkoušela ve třídě P3. Testování pracovních listů probíhalo během dvou dvouhodinových bloků na začátku školního roku 2021/2022. Během prvního bloku se stihl úvod do vývojového prostředí a základy práce s micro:bitem (propojení s počítačem apod.). Následovalo řešení úloh z pracovního listu „Osová souměrnost“, kdy žáci samostatně přicházeli na řešení jednotlivých zadání úloh. V průběhu jejich práce se vyskytly dva problémy: prvním problémem bylo správné uvědomění si zadání zadaných úloh – bylo žákům třeba připomenout, co to je osová souměrnost, a druhým vyskytnutým problémem byla „jazyková bariéra“ – žáci během prázdnin pozapomněli klíčová slova z JavaScriptu. Žákům se tedy ukázalo přepínání mezi jazyky, čímž si mohli pomoci, když si nevěděli rady. Také si mohli rozkliknout knihovnu klíčových slov.

Na druhé hodině prvního bloku žáci dokončovali řešení úloh z pracovního listu „Osová souměrnost“ a začínali s vypracováváním pracovního listu s názvem „Středová souměrnost“. Během druhého bloku se už nevyskytl žádný z výše popsanych problémů.

5.4 Vybraná žákovská řešení úloh z pracovních listů

V této části jsou ukázána vybraná žákovská řešení úloh z pracovních listů. Jsou zde zmíněny nastalé obtíže a nejčastější způsoby řešení úloh žáky.

Úvodní hodina

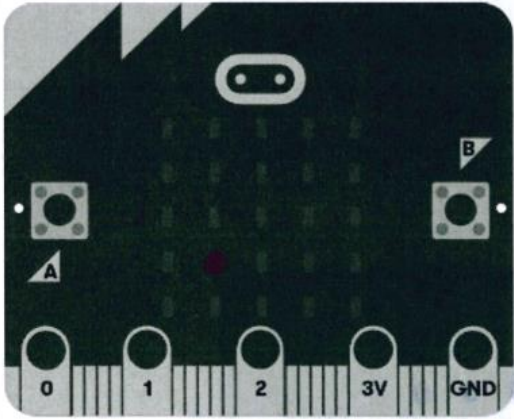
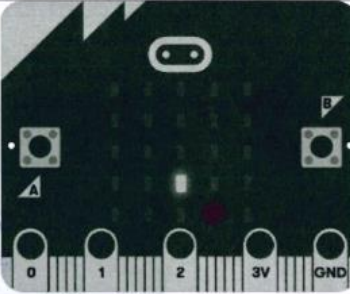
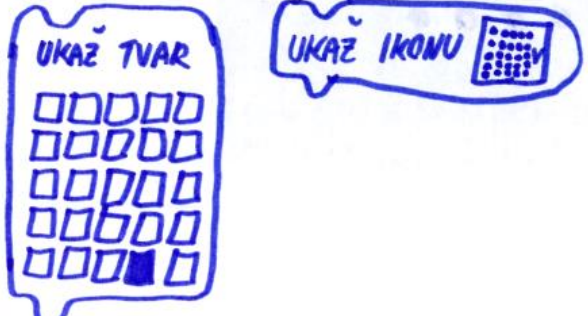
S vypracováváním úloh v pracovním listě „Úvodní hodina“ neměl žádný žák obtíže.

1. úloha

1. úloha
Jak funguje číslování řádků a sloupců?
$0 = 4$

Obrázek 67: Vybrané žákovské řešení 1. úlohy z pracovního listu Úvodní hodina

2. úloha

2. úloha	Vzorové řešení
Rozsviť bod na souřadnicích (1, 3)	Rozsviť bod na souřadnicích (3, 4)
Označ si, která dioda se rozsvítí?	Jak vypadá kód:
	
	Jak se to zobrazí:
	
Existuje více možností, jak rozsvítit bod. Dokážeš najít nějaký další způsob?	
	

Obrázek 68: Vybrané žákovské řešení 2. úlohy z pracovního listu Úvodní hodina

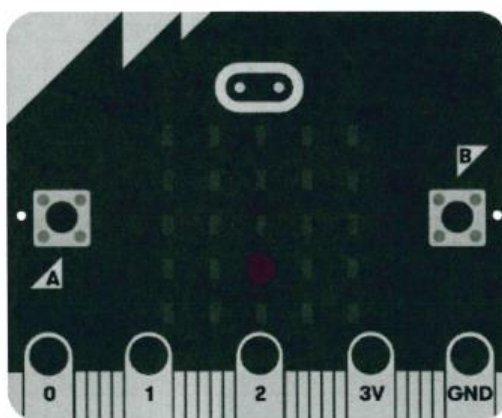
Na obrázku č. 68 si lze všimnout chyby ve vzorovém řešení úlohy č. 2 v pracovním listu s názvem „Úvodní hodina“. Autor vypracovávaného pracovního listu znázorněného na obr. 68 nedopatřením ponechanou chybu ve vzorovém řešení objevil, chybné řešení přepsal a vyznačil si i v obrázku micro:bitu správné vzorové řešení.

Na otázku, zda existují další způsoby řešení úlohy, se také vyskytovala odpověď „Rozsvít' x(1) y(3) s jasem (155)“.

3. úloha

Rozsvít bod na souřadnicích (2, 3) s jasem 150.

Označ si, která dioda se rozsvítí?

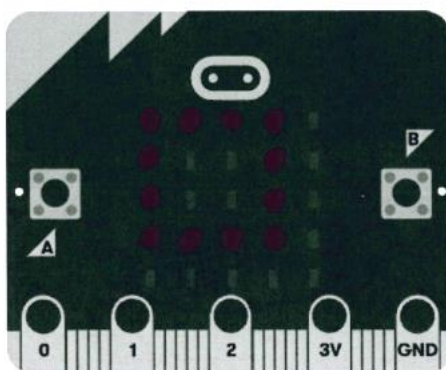


Obrázek 69: Vybrané žákovské řešení 3. úlohy z pracovního listu Úvodní hodina

4. úloha

Po stisknutí tlačítka A rozsvít obvod čtverce se stranou 3 jednotky* libovolně umístěný na Micro:bit
 * jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami

Označ si, která dioda se rozsvítí?



Obrázek 70: Vybrané žákovské řešení 4. úlohy z pracovního listu Úvodní hodina

5.4.1 Vybraná žákovská řešení úloh z pracovních listů určených pro žáky ZŠ

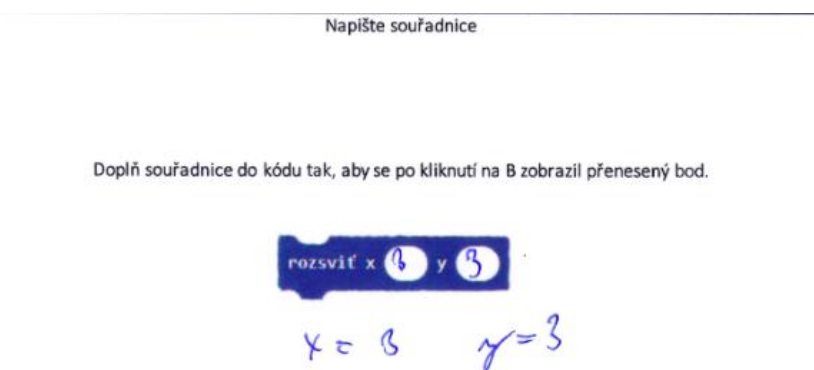
Většina žáků nevyužila možnosti vyznačování řešení přímo v obrázcích prázdného micro:bitu, nebo micro:bitu s vysvícenými LED diodami představujícími zadané body či objekty, které jsou znázorněny v jednotlivých pracovních listech. Spousta žáků také nenarýsovala konstrukční řešení geometrického problému. Žáci, kteří využili „převedení problému na geometrický“, řešení často pouze načrtli, a to pravděpodobně jen v případě, když si nebyli jisti, jak se úloha řeší, či jak funguje dané shodné zobrazení.

Občas někteří žáci zapomněli na druhou stranu pracovního listu, a tak se tedy stalo, že poslední z úloh v pracovním listě nebyla vyřešena, a událo se tomu i přes upozornění, že je pracovní list oboustranný.

Osová souměrnost

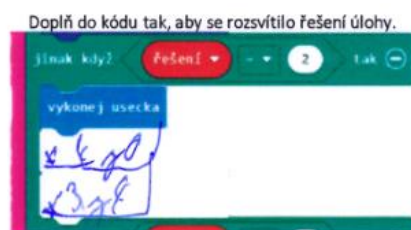
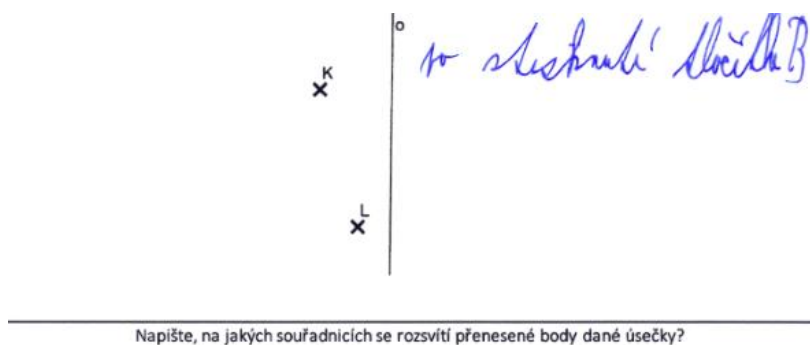
Během sestavování programů se u žáků vyskytla metoda „pokus – omyl“ při vysvěcování LED diod micro:bitu. Žáci si také pletli osu x s osou y .

1. úloha



Obrázek 71: Vybrané žákovské řešení 1. úlohy z pracovního listu *Osová souměrnost*

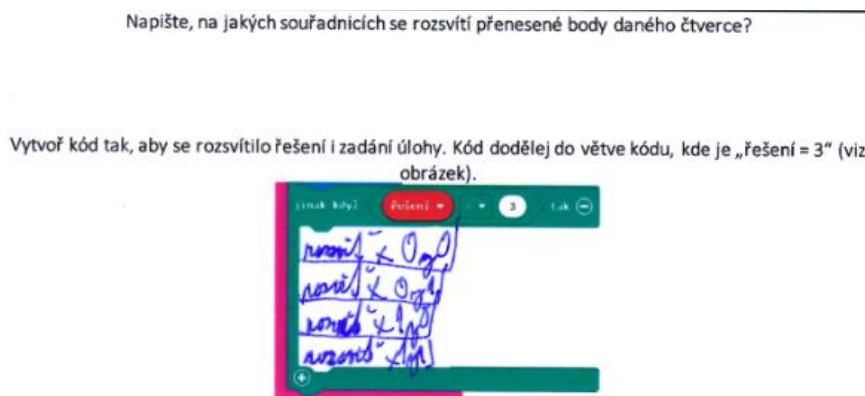
2. úloha



Obrázek 72: Vybrané žákovské řešení 2. úlohy z pracovního listu *Osová souměrnost*

Na obrázku č. 72 lze sledovat, že si vybraný žák upravil v pracovním listu zadání úlohy.

3. úloha



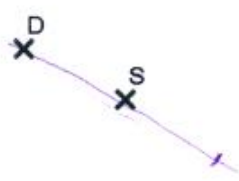
Obrázek 73: Vybrané žákovské řešení 3. úlohy z pracovního listu *Osová souměrnost*

Středová souměrnost

Žáci při vypracovávání úloh z pracovního listu „Středová souměrnost“ již neměli problém s rozsvěcováním řešení, nepoužívali již metodu „pokus – omyl“. Dokonce zjistili, že si práci mohou usnadnit, a to tím, že pokud podrželi myš na diodě v simulaci ve vývojovém prostředí, prostředí jim samo ukáže, jaké souřadnice příslušná dioda má.

1. úloha

Převodění problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí přenesený bod?

Doplň souřadnice do kódu tak, aby se po kliknutí na B zobrazil přenesený bod.

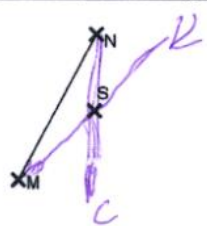
```
rozsviť x 4 y 3
```

Obrázek 74: Vybrané žákovské řešení 1. úlohy z pracovního listu *Středová souměrnost*

Na obrázku č. 74 vidíme, že autor vyplněného pracovního listu pouze načrtl geometrické řešení a že zapomněl odpovědět na otázku, na jakých souřadnicích se rozsvítí výsledný bod. Někteří žáci zapomněli zapsat na předpřipravené volné místo odpověď na otázku, ale ve znázorněném bloku byl výsledek vždy zapsán.

2. úloha

Převodění problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí přenesené body dané úsečky?

Doplň do kódu tak, aby se rozsvítilo řešení úlohy.

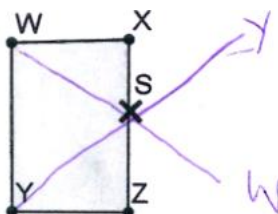
Jinak kdy?

```
vykonej úsečka  
rozsviť x 4 y 0  
rozsviť x 2 y 4
```

Obrázek 75: Vybrané žákovské řešení 2. úlohy z pracovního listu *Středová souměrnost*

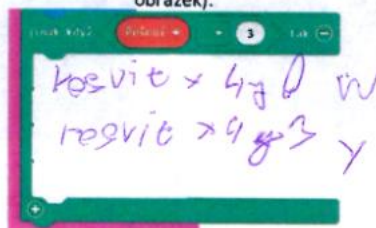
Na obrázku č. 75 se vyskytují již výše zmíněné jevy, tj. pouze načrtnuté řešení a chybějící odpověď na otázku. Ve spodní části obrázku vidíme v zobrazeném programu příkaz pouze napsaný, na druhou stranu někteří žáci odpovídající bloky do pracovních listů kreslili.

3. úloha



Napište, na jakých souřadnicích se rozsvítí přenesené body daného obdélníka?

Vytvoř kód tak, aby se rozsvítilo řešení i zadání úlohy. Kód dodělej do větve kódu, kde je „řešení = 3“ (viz obrázek).



Obrázek 76: Vybrané žákovské řešení 3. úlohy z pracovního listu *Středová souměrnost*

Na obrázku č. 76 je vidět zápis neúplného řešení úlohy. Občas se stalo, že žáci do pracovního listu napsali jen část řešení, a to i přes to, že program měli sestavený tak, že ukazoval celé řešení.

Posunutí

V pracovním listě Posunutí žáci volili mnoho různých způsobů řešení. Jedním z nich bylo vysvěcování tvaru, který se každé 3 s měnil, dalším způsobem bylo užití příkazu „rozsvit' x () y ()“ a posledním využitým způsobem řešení úlohy bylo zakomponování proměnných a cyklů.

V rámci vypracovávání úloh v tomto pracovním listě využil pouze jeden žák možnosti geometrického znázornění řešení problému.

1. úloha

Jak bude vypadat výsledný kód, pokud chceme použít proměnné?

při startu
nastav y na -1

každé 3 sekundy
změň y o 1

Opakuj stále
zhasni $x(0,1,2,3,4)$ $y(y+1)$
rozsviť $x(0,1,2,3,4)$ $y(y)$

Obrázek 77: Vybrané žákovské řešení 1. úlohy z pracovního listu Posunutí

Na obrázku č. 77 je ukázáno řešení s využitím proměnných a cyklů. Žák, který je autorem zobrazeného řešení, se snažil o úsporný zápis řešení úlohy.

2. úloha

Jak bude vypadat výsledný kód, pokud chceme použít proměnné?

při startu
nastav x na -1

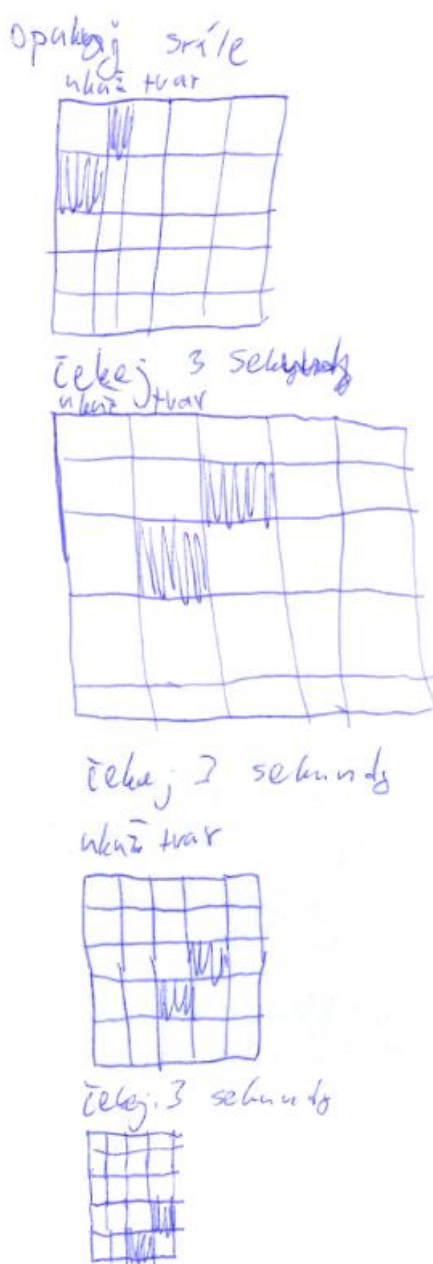
každé 3 sekundy
změň x o 1

Opakuj stále
zhasni $x(x-1)$ $y(0,...4)$
rozsviť $x(x)$ $y(0,...4)$

Obrázek 78: Vybrané žákovské řešení 2. úlohy z pracovního listu Posunutí

Na obrázcích č. 77 a č. 78 můžeme pozorovat, že se příslušný žák nezamýšlel nad tím, co se stane, když jedna ze souřadnic bude větší než 5. Předpokládal nejspíše, že když úsečka zmizí mimo matici LED diod, program skončí.

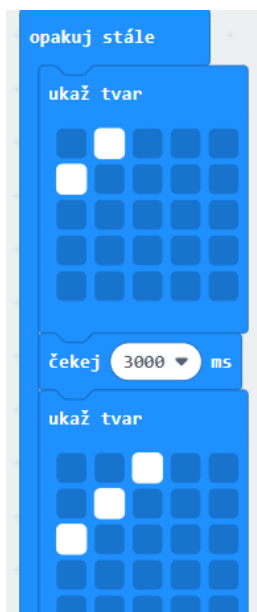
3. úloha



Obrázek 79: Vybrané žákovské řešení 3. úlohy z pracovního listu Posunutí

Na obrázku č. 79 vidíme, že se řešení nevešlo do kolonky pro odpověď, a zároveň, že se žák pokusil o jiný způsob řešení. V tomto uvedeném způsobu řešení také vidíme, že žák promyslel, co se má stát, když úsečka opustí systém souřadnic micro:bitu.

Častou chybou při řešení této úlohy bylo, že žáci zaměnili úsečku za přímku a nerozsvěcovali postupně pouze dvě diody, ale celý diagonální řádek. Jiní žáci také úlohu pochopili mylně tak, že nejenže chybně rozsvěcovali celý diagonální řádek, na němž ležely dvě diody představující krajní body úsečky, ale že rozsvěcovali každý diagonální řádek v matici LED diod.



Obrázek 80: Ukázka řešení úlohy číslo 3 z pracovního listu Posunutí

Otočení

1. úloha

Napište, na jakých souřadnicích se rozsvítí otočený bod?

$$x=4 \quad y=0$$

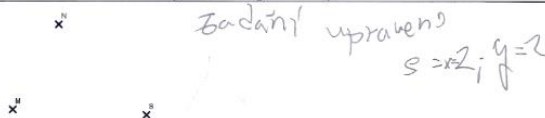
Doplň souřadnice do kódu tak, aby se po kliknutí na B zobrazil otočený bod.



Obrázek 81: Vybrané žákovské řešení 1. úlohy z pracovního listu Otočení

2. úloha

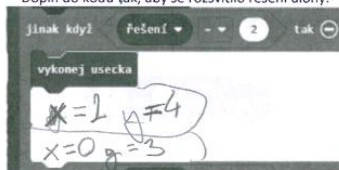
Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí otočené body dané úsečky?

$$x=2 \quad y=4 \quad x=0 \quad y=3$$

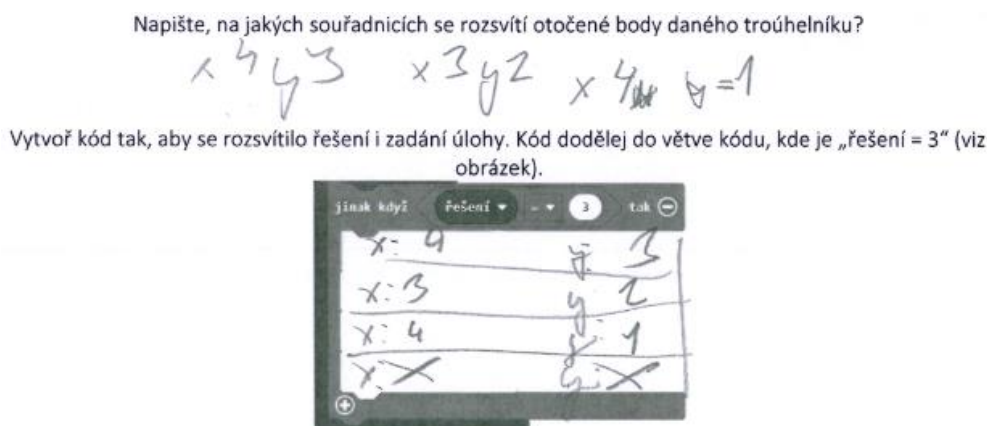
Doplň do kódu tak, aby se rozsvítlo řešení úlohy.



Obrázek 82: Vybrané žákovské řešení 2. úlohy z pracovního listu Otočení

Obrázek č. 82 ukazuje úpravu zadání úlohy kvůli nalezené chybě. Někteří žáci si poznámku o upraveném zadání nepsali.

3. úloha



Obrázek 83: Vybrané žákovské řešení 3. úlohy z pracovního listu Otočení

5.4.2 Vybraná žákovská řešení úloh z pracovních listů určených pro žáky SŠ

Ani žáci na střední škole nevyužili možnost vyznačení bodů, resp. objektů ve znázorněném obrázku micro:bitu. Častými problémy, které se při vypracovávání úloh zařazených v pracovních listech na střední škole vyskytly, byly, že žáci zapomněli základní principy a vlastnosti shodných zobrazení, anebo potřebné příkazy či klíčová slova.

Každý žák střední školy využíval proměnných, cyklů a podmínek. Během testování se vyskytly zhruba tři mírně odlišné přístupy, žáci pracovali ve skupinách po třech až po čtyřech.

1. úloha

Kód v JavaScript

Jak vypadá kód v JavaScript?
Zapiš si, jak vypadají jednotlivé příkazy:

```

input.onButtonPressed (Button.A, function () {
    ClearStuff()
    RandomGen()
    GenerateCopy()
})
function ClearStuff () {
    for (let i = 0; i <= 4; i++)
        for (let j = 0; j <= 4; j++) {
            led.unplot (i, j)
        }
    for (let i = 0; i <= 4; i++) {
        led.plotBrightness (2, i, 100)
    }
}
function RandomGen () {
    let random = randint (0, 3)
    for (let i = 0; i <= 1; i++) {
        led.plot (0, random+i)
        led.plot (4, random+i)
    }
}
            
```

```

function GenerateCopy () {
    for (let x = 0; x <= 1; x++) {
        for (let y = 0; y <= 4; y++) {
            if (led.point (x, y)) {
                led.plot (4-x, y)
            }
        }
    }
    for (let i = 0; i <= 4; i++) {
        led.plotBrightness (2, i, 100)
    }
}
            
```

Obrázek 84: Vybrané žákovské řešení 1. úlohy z pracovního listu Osová souměrnost

Pro některé žáky byl velkou překážkou malý prostor určený v pracovním listě pro zápis kódu. Tento problém jsme vyřešili tím, že někteří žáci mi poslali svůj kód na e-mail.

2. úloha



Obrázek 85: Vybrané žákovské řešení 1. úlohy z pracovního listu Osová souměrnost

5.5 Tvorba dotazníku

Tato podkapitola je věnována popisu tvorby dotazníku a uskutečněnému dotazníkovému šetření. Pro tuto diplomovou práci byly vytvořeny dvě verze jednoho a téhož dotazníku, jedna jako tištěná forma a druhá jako online verze (viz https://docs.google.com/forms/d/e/1FAIpQLSeKMiBT_nkKgS82-NZx6ZLKl1FqE89XXvVopttfPG_ni5vRDw/viewform?usp=sf_link). Online verze dotazníku byla vytvořena především z důvodu případného přechodu prezenční formy výuky na distanční formu výuky.

Dotazník byl vytvořen za účelem získání zpětné vazby od žáků ve spojitosti s jejich postřehy týkajícími se srozumitelnosti zadání úloh uvedených v pracovních listech, úrovně obtížnosti zadaných úloh, náročnosti hledání správných řešení z hlediska geometrie, ale i z hlediska sestavených programů pro micro:bit. Je počítáno s tím, že

pracovní listy budou na základě získané zpětné vazby od žáků upraveny, případně doplněny, aby odpovídaly úrovni a potřebám žáků příslušného stupně školy.

5.5.1 Základní principy tvorby dotazníků

Při tvorbě dotazníku je v první řadě důležité zamyslet se nad cílem jeho vytvoření. Nad tím, co pomocí něj chceme zjišťovat. Je tedy velmi podstatné položit si základní otázku: „Jaký je cíl dotazníku?“ nebo „Co chceme zjišťovat?“. ([3], str. 99)

Dotazník by měl mít pevnou strukturu. Skládá se zpravidla ze tří základních částí. První část se nazývá vstupní, v ní se nachází hlavička dotazníku (název instituce, pod kterou je dotazník vytvořen, jméno autora). Dále se v této části vysvětlují cíle dotazníku a jaký význam mají odpovědi respondentů (osoby, které vyplňují dotazník) pro autora dotazníku, případně, jak budou získané odpovědi dále využity. A poslední součástí vstupní části dotazníku jsou většinou pokyny k jeho vyplnění. Druhá část dotazníku obsahuje vlastní otázky. A poslední, třetí částí dotazníku bývá poděkování respondentům za vyplnění dotazníku. ([3], str. 99 – 100)

Tvoření otázek dotazníku má devět základních pravidel:

1. Jasně formulace otázek.
2. Široké znění otázky vede k otevřené odpovědi.
3. Vyhýbat se výrazům „někdy“, „několik“ nebo „obyčejně“ apod. – každý respondent může uvedené výrazy vnímat jinak.
4. Vyhýbat se dvojitým otázkám (tzn. vyhýbat se spojení „a“).
5. Uvádět otázky, na které lze odpovědět.
6. Používat smysluplné otázky.
7. Používat jednoduché otázky.
8. Vyhýbat se záporným výrazům.
9. Klást otázky bez ovlivnění autorem. ([3], str. 100 – 101)

Hlavní cíle sestaveného dotazníku

Cílem sestaveného dotazníku je získání zpětné vazby k vytvořeným pracovním listům a úlohám, které jsou v nich obsaženy. Poté na základě získané zpětné vazby pracovní listy, resp. úlohy vložené v nich upravit, případně doplnit, nebo pozměnit.

Základní zjišťovaný problém je „použitelnost pracovních listů“. V první řadě je důležité zjistit, zda jsou všechna uvedená zadání úloh pro většinu žáků srozumitelná, a dále, zda

jsou všechny úlohy odpovídajících obtížností vzhledem k věku a dosavadním znalostem žáků.

5.5.2 Pilotní dotazníkové šetření

V rámci kroužku robotiky, který jsem vedla pro zájemce z řad žáků 2. stupně ZŠ Vrchlického v Liberci a kde jsem také zkoušela zařadit programování s micro:bity, proběhlo pilotní vyplnění dotazníku za účelem překontrolování validity pokládaných otázek.

Jednalo se o kroužek robotiky, který navštěvovaly děti od 5. třídy po 8. třídu. Vzhledem k tomu, že pouze děti z 8. třídy rozuměly pojmům jako osová a středová souměrnost apod., musela jsem dětem z kroužku nejprve vysvětlit, co jednotlivé pojmy znamenají a co se od nich vlastně očekává. Jednalo se o 7 dětí, přičemž na kroužek docházelo šest chlapců a pouze jedna dívka.

5.5.3 Zpětná vazba

Zpětnou vazbu jsem získávala od žáků základní školy pomocí odpovědí na otázky položené v dotazníku. Zpětnou vazbu od žáků střední školy jsem získala rovněž z odpovědí na otázky položené v dotazníku, ale navíc jsem získala ještě i zpětnou vazbu od pana učitele zmíněné střední školy.

Vyhodnocení dotazníků

Tabulka 2: Základní údaje

Škola	ZŠ	SŠ
	37	32
Pohlaví	Chlapec	Dívka
	53	14

Hodnoty zapsané v tabulce č. 2 ukazují, že dotazník vyplnilo více žáků ze základní školy než ze střední školy, což může být způsobeno hned několika faktory. Prvním faktorem je fakt, že pan učitel ze SŠ zapomněl po ukončení vypracování každého pracovního listu žákům připomenout vyplnění dotazníku, a druhým faktorem je skutečnost, že v jedné třídě na ZŠ, ve které testování pracovních listů probíhalo, vypracovali žáci všechny připravené pracovní listy oproti žákům SŠ, kde každý žák vypracoval v průměru pouze dva pracovní listy.

Pracovní listy testovalo více chlapců než dívek, viz tabulka č. 2. Pouze dva respondenti nevyužili možnost identifikace pohlaví.



Graf 1: Počty vyplněných pracovních listů v jednotlivých testovaných třídách

V grafu č. 1 vidíme počty pracovních listů, které žáci v příslušných třídách vypracovali a odevzdali, a u žáků ZŠ tedy také počty vyplněných dotazníků se zpětnou vazbou k pracovním listům. Třída 8.B je v uvedených počtech zastoupena nejvíce, protože její žáci vypracovali všechny pracovní listy a ke každému pracovnímu listu poskytli zpětnou vazbu formou odpovědí na otázky dotazníku. Žáci zbylých tříd, ve kterých testování pracovních listů probíhalo, vypracovali pouze jeden nebo dva pracovní listy.

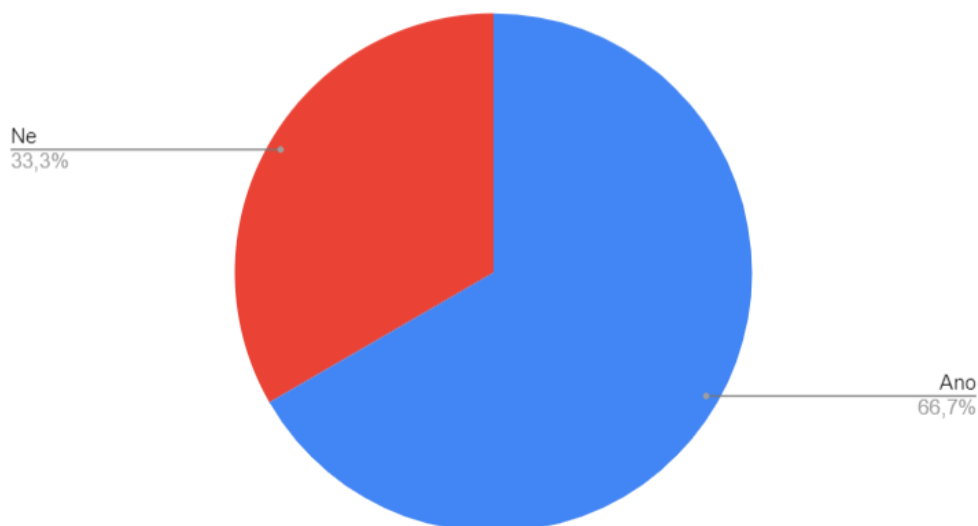
Tabulka 3: Počty vypracovaných pracovních listů pro jednotlivá témata z geometrie

Název pracovního listu:	Otočení	Posunutí	Osová souměrnost	Středová souměrnost	Kombinované
Počet vypracovaných pracovních listů	21	22	14	9	3

Pro oba stupně škol byly vytvořeny pracovní listy pro stejná témata z geometrie. Úlohy se lišily pouze obtížností a svým typem. Úlohy v pracovních listech vytvořené pro žáky základní školy patří mezi tzv. polohové úlohy, tj. je známá poloha zadaných geometrických

objektů v rovině. Na druhou stranu úlohy vyskytující se v pracovních listech pro žáky střední školy spadají mezi tzv. nepolohové úlohy, tj. není známá poloha zadaných geometrických objektů v rovině. Nejvíce vyzkoušených pracovních listů je pro témata Otočení a Posunutí, což může být způsobeno tím, že se na střední škole zadávaly především tyto dva pracovní listy. Také si myslím, že někteří žáci vyplňovali dotazník pro všechny pracovní listy najednou a ne zvlášť pro jednotlivé pracovní listy, proto je v tabulce č. 3 uvedena kolonka kombinované. Bylo vypracováno mnohem více pracovních listů, než vyplývá z odevzdaných dotazníků. To může být i důvodem, proč procentuální zastoupení žáků ze střední školy je v odpovědích na otázky dotazníku menší než procentuální zastoupení žáků ze základní školy.

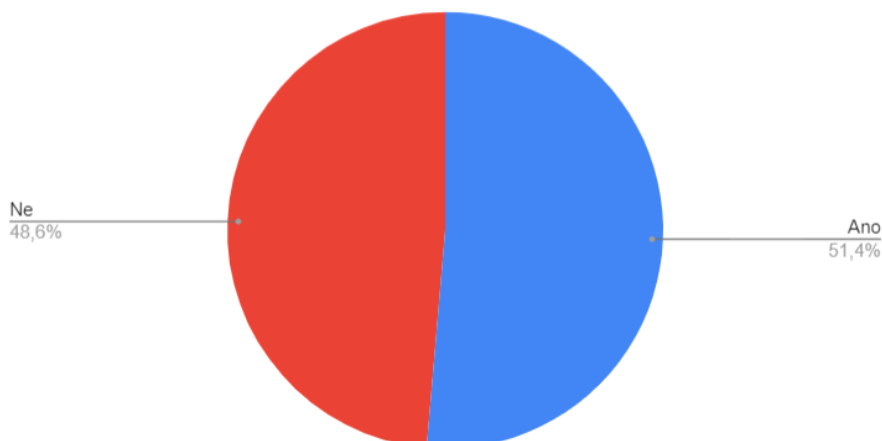
Rozuměl jsi zadání všech úloh v pracovním listu?



Graf 2: Procentuální zastoupení (ne)porozumění zadání úloh žáky

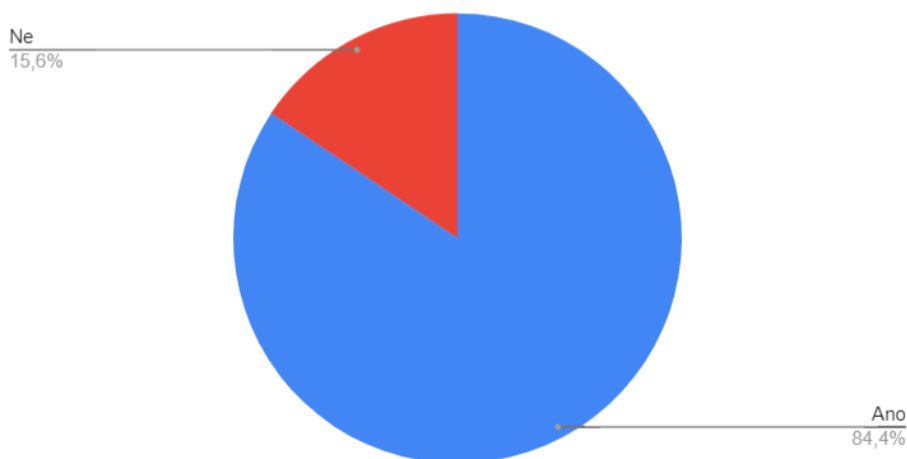
Pouze jedna třetina žáků základní a střední školy, kteří vyplňovali dotazník, neporozuměla zadání některé z úloh v pracovních listech. Mezi častější důvody, které žáci uvedli, patří nepochopení úlohy. Další důvody se nevztahovaly k neporozumění zadáním, ale k problémům s nalezením řešení některé úlohy. Někteří žáci uváděli, že měli problém se sestavením algoritmu a s následným vytvořením programu v nějakém z nabízených jazyků. Tři žáci poukázali na „nefunkčnost“ webového vývojového prostředí makecode.microbit.org, která však nesouvisí s neporozuměním zadání úloh v pracovních listech.

Rozuměl jsi zadání všech úloh v pracovním listu? (Žáci základní školy)



Graf 3: Procentuální zastoupení (ne)porozumění zadání úloh žáky základní školy

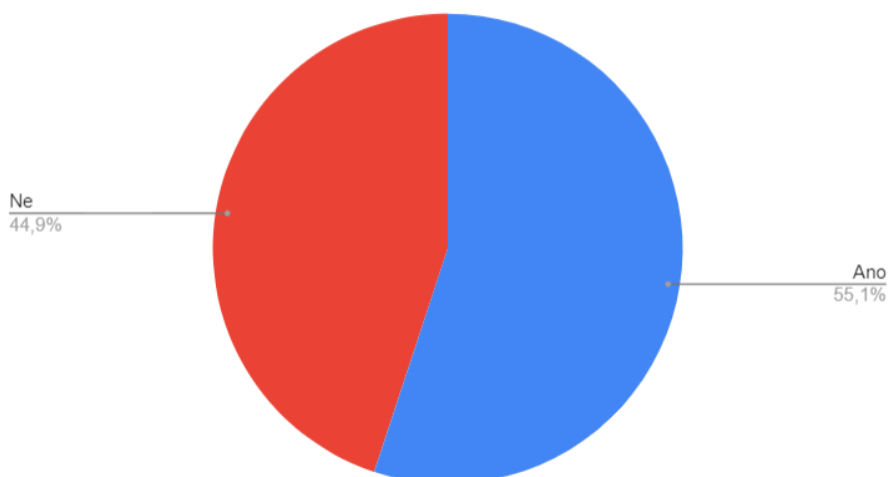
Rozuměl jsi zadání všech úloh v pracovním listu? (Žáci střední školy)



Graf 4: Procentuální zastoupení (ne)porozumění zadání úloh žáky střední školy

Z porovnání procentuálních hodnot uvedených v koláčových grafech č. 3 a č. 4 můžeme pozorovat, že žáci střední školy měli menší obtíže s porozuměním zadání úloh i přes to, že zadání úloh v pracovních listech určených pro žáky střední školy byla komplikovanější než zadání úloh v pracovních listech určených pro žáky základní školy. Téměř polovina žáků na základní škole měla problém s porozuměním zadání některé z úloh v pracovních listech.

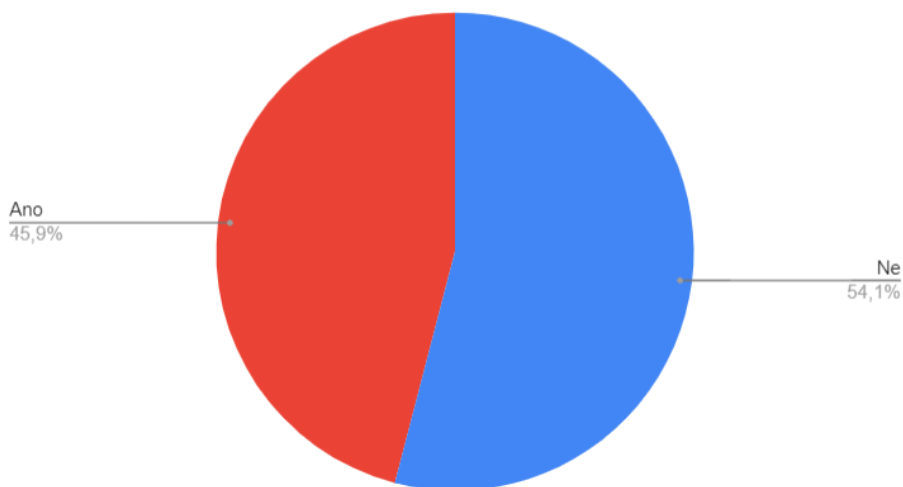
Přišla ti nějaká z úloh obtížná?



Graf 5: Procentuální zastoupení vyjádření žáků k obtížnostem zadáných úloh

Jako nejtěžší úlohu žáci zvolili úlohu č.3 z pracovního listu Posunutí, který byl pro oba dva stupně škol stejný. Mezi poznamenané důvody patřilo: mnoho bodů a ne příliš jasný směr posunutí. Osobně se domnívám, že bylo na této úloze nejtěžší sestavení samotného programu.

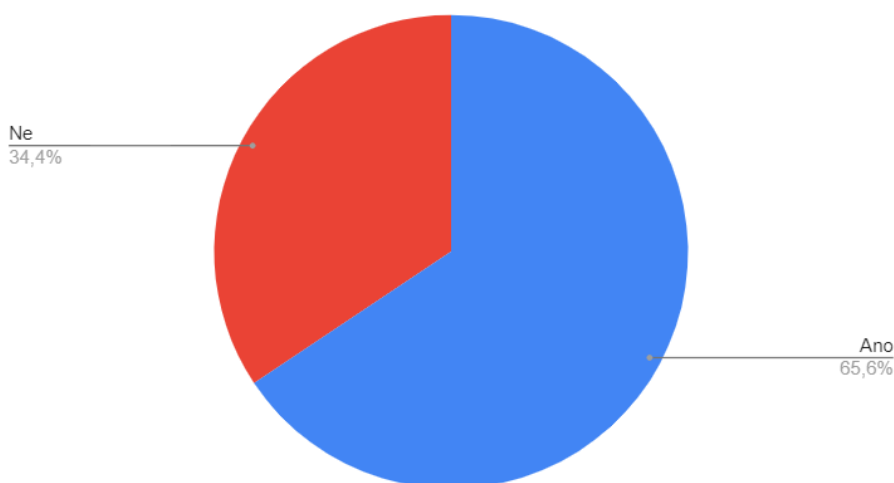
Přišla ti nějaká z úloh obtížná? (Žáci základní školy)



Graf 6: Procentuální zastoupení vyjádření žáků ZŠ k obtížnostem zadáných úloh

Koláčový graf č. 6 ukazuje, že téměř polovina žáků ZŠ vnímala některou z úloh v pracovních listech jako složitou. Zajímavým zjištěním bylo, že i přestože úlohu č. 3 z pracovního listu Posunutí vnímali žáci jako složitou, označili tuto úlohu i jako nejvíce oblíbenou. Jako důvody oblíbenosti zvolené úlohy žáky základní školy byly uváděny její neobvyklost oproti ostatním úlohám zařazeným v témže pracovním listě, což je kontrastu s tím, že tutéž úlohu označili jako obtížnou.

Přišla ti nějaká z úloh obtížná? (Žáci střední školy)



Graf 7: Procentuální zastoupení vyjádření žáků SŠ k obtížnostem zadaných úloh

Z koláčového grafu č. 7 vidíme, že jedna třetina žáků střední školy nevnímala úlohy v pracovních listech jako složité. I žáci střední školy uvedli úlohu č. 3 v pracovním listě „Posunutí“ jako nejsložitější. Z uvedených odpovědí vyplývá, že se žákům střední školy nejvíce líbila úloha č. 1 z pracovního listu na téma „Posunutí“. Častými důvody oblíbenosti této úlohy žáky střední školy byla jednoduchost úlohy a její zajímavost.

Poslední otázkou dotazníku byla otázka „Co bys změnil/a na pracovním listu?“ Dvacet žáků SPŠSE a VOŠ v Liberci by na pracovních listech neměnilo nic. Důležité jsou ale konstruktivní připomínky, ze kterých mohu čerpat při úpravě pracovních listů a také při tvorbě dalších pracovních listů nebo testů ve své učitelské praxi. To osobně vnímám jako velký přínos pro mé budoucí povolání.

Připomínky od žáků, které mě nejvíce oslovily, jsou například více úloh v pracovním listu, větší různorodost úloh v rámci jednoho pracovního listu nebo připsání obtížností k jednotlivým úlohám.

Ke zdokonalení pracovních listů mě inspirovaly tyto dvě připomínky: zvětšení písma na pracovních listech a více místa pro zapsání kódu.

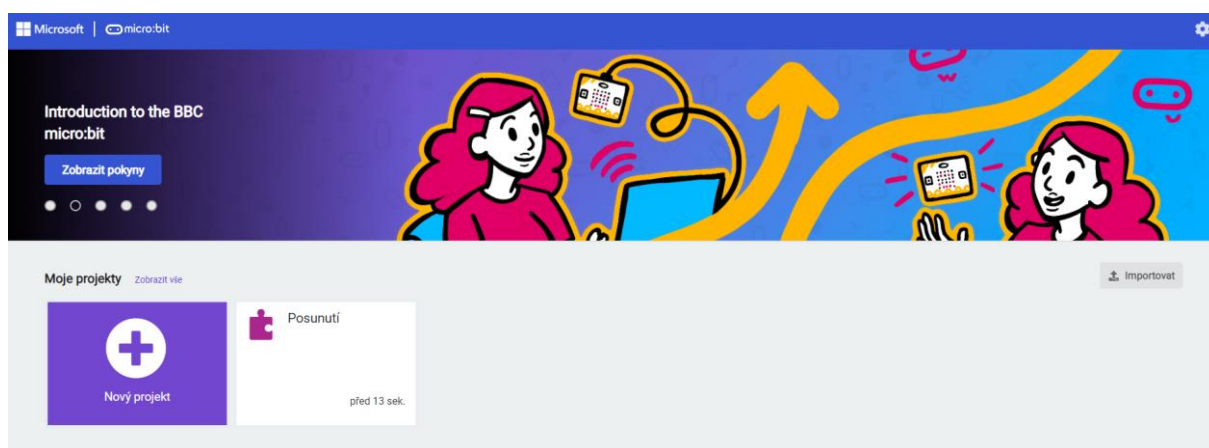
K zamyšlení je také připomínka ke psaní kódu na papír vůbec, možná by bylo vhodnější najít lepší způsob zaznamenávání kódu, např. poslání kódu e-mailem, vyfocení kódu apod.

5.6 Změny a úpravy pracovních listů

Na základě zpětné vazby získané od pana učitele ze SŠ jsem k pracovním listům vytvořila stručnou metodickou příručku pro učitele.

Dle připomínek žáků k pracovním listům by bylo vhodné pracovní list tisknout pouze ve formátu A4. *Poznámka autorky:* Ve třídě 8.B žáci obdrželi pracovní list ve formátu A5.

Pracovní listy byly také lehce poupraveny v tom smyslu, že z nich byla odstraněna část „Odkaz“, a to především kvůli mé původní mylné představě, že se všichni žáci dostanou na stejný odkaz. Projekty, které lze po otevření internetové stránky makecode.microbit.org otevřít, nejsou totiž na každém počítači stejné. Jedná se o jakési „cookies soubory“, které se ukládají do paměti počítače a usnadňují tak práci na některých stránkách, proto projekt nemusíme opakovaně stahovat a nahrávat opětovně do vývojového prostředí. Tyto soubory ale má pouze počítač, na kterém se pracuje, a žádný jiný.



Obrázek 86: Micro:bit makecode – projekty

Během testování pracovních listů se v zadáních úloh narazilo i na „technické chyby“. Nebylo možné vysvětlit řešení úlohy např. v pracovním listě určeném pro žáky základní školy na téma „Otočení“, kdy v úloze č. 2 byl špatně vysvěcený střed otočení.

Mimo jiné byly opraveny drobnější chyby s nedostatečným volným místem, aby se při rýsování veškeré pomocné konstrukce vešly na vynechaná místa.

5.7 Metodická příručka

Metodické pomůcky jsou určeny výhradně učitelům. Slouží k užití správné metody učitelem během vyučovacího procesu. Mezi tyto pomůcky také patří mimo jiné metodické příručky. ([11], str. 206)

Smyslem metodických listů/příruček je jejich praktické využití jako komplexní přípravy na vyučovací hodinu pro širokou vrstvu pedagogů. ([11], str. 207)

Metodická příručka by měla obsahovat didaktickou část zahrnující sledované cíle, pomůcky a nástroje, motivační složku, pracovní postupy, mezipředmětové vztahy, organizační doporučení. ([11], str. 207)

5.7.1 Sledované cíle

Jedním z cílů vytvořených pracovních listů je názorná ukázka propojení matematiky a informatiky. Je třeba žákům ukázat fakt, že je informatika s matematikou velmi úzce propojena a že je potřeba k úspěšnému vyřešení některých úloh z informatiky znát a ovládat některé oblasti matematiky (např. výroková logika apod.).

Mimo jiné navržené pracovní listy pro žáky ZŠ sledují cíle z Rámcového vzdělávacího programu pro základní školy dle nové revize z roku 2021, kdy proběhla revize především v oblasti informatiky. U pracovních listů vytvořených pro žáky střední školy se jedná o cíle z Rámcového vzdělávacího programu pro střední školy se zaměřením na informační technologie (RVP 18-20-M/01).

Základní škola:

- Matematika: (geometrie v rovině a v prostoru)
 - **M-9-3-02** žák charakterizuje a třídí základní rovinné útvary [12]
 - **M-9-3-08** žák načrtne a sestojí obraz rovinného útvaru ve středové a osové souměrnosti, určí osově a středově souměrný útvar [12]
- Informatika: (algoritmizace a programování)
 - **I-5-2-02** žák popíše jednoduchý problém, navrhne a popíše jednotlivé kroky jeho řešení [12]
 - **I-5-2-03** v blokově orientovaném programovacím jazyce žák sestaví program; rozpozná opakující se vzory, používá opakování a připravené podprogramy [12]
 - **I-9-2-05** v blokově orientovaném programovacím jazyce žák vytvoří přehledný program s ohledem na jeho možné důsledky a svou odpovědnost za něj; program vyzkouší a opraví v něm případné chyby; používá opakování, větvení programu, proměnné [12]
- Kompetence:
 - Digitální kompetence

- žák chápe význam digitálních technologií pro lidskou společnost, seznamuje se s novými technologiemi, kriticky hodnotí jejich přínosy a reflektuje rizika jejich využívání [12]

Střední škola

V Rámcovém vzdělávacím programu pro střední školy nejsou cíle vymezeny tak, jako v Rámcovém vzdělávacím programu pro základní školy.

- Matematika:
 - shodná a podobná zobrazení (Planimetrie) [13]
 - vektory (Analytická geometrie v rovině) [13]
 - přímka a její analytické vyjádření (Analytická geometrie v rovině)
 - žák řeší analyticky polohové a metrické vztahy bodů a přímek;
 - žák užívá různá analytická vyjádření přímky; [13]
- Informatika:
 - algoritmizace (Práce s počítačem, operační systém, soubory, adresářová struktura, souhrnné cíle)
 - žák ovládá principy algoritmizace úloh a sestavuje algoritmy řešení konkrétních úloh (dekompozice úlohy na jednotlivé elementárnější činnosti za použití přiměřené míry abstrakce); [13]
- Kompetence:
 - Kompetence využívat prostředky informačních a komunikačních technologií a pracovat s informacemi: (klíčové kompetence)
 - pracovat s osobním počítačem a dalšími prostředky informačních a komunikačních technologií; [13]
 - Programovat a vyvíjet uživatelská, databázová a webová řešení: (odborné kompetence)
 - algoritmizovat úlohy a tvořit aplikace v některém vývojovém prostředí. [13]
 - Matematické kompetence: (klíčové kompetence)
 - nacházet vztahy mezi jevy a předměty při řešení praktických úkolů, umět je vymezit, popsat a správně využít pro dané řešení; [13]
 - číst a vytvářet různé formy grafického znázornění (tabulky, diagramy, grafy, schémata apod.); [13]

- aplikovat znalosti o základních tvarech předmětů a jejich vzájemné poloze v rovině i prostoru; [13]
- efektivně aplikovat matematické postupy při řešení různých praktických úkolů v běžných situacích. [13]

5.7.2 Pomůcky a nástroje

K úspěšnému vyřešení úloh zařazených ve vytvořených pracovních listech jsou potřeba micro:bit, USB kabel na propojení micro:bitu a počítače, počítač/notebook, připojení k internetu, rýsovací a psací pomůcky.

5.7.3 Motivace

Motivací pro žáky by mohla být ukázka úzkého vztahu informatiky a matematiky, tj. ukázka toho, že za každou internetovou stránkou, hrou nebo robotem je nějaký skrytý matematický aparát. Například u programování micro:bitu to je znalost kartézského souřadnicového systému a následné umísťování bodů v něm, správné používání logických spojek a z toho plynoucí užívání složených výroků apod.

Úzký vztah je v pracovních listech opřen o vybrané matematické problémy z oblasti geometrie, které žáci musí rozdělit na dílčí části a systematicky pak programovat řešení těchto dílčích částí. Další motivací je tedy připravenost na rozklad mnohem složitějších problémů na ty jednodušší, a to platí někdy i v běžném životě pro snazší systematické řešení jednodušších problémů běžného života.

5.7.4 Pracovní postupy

Prvním krokem pracovního postupu by mělo být rozebrání úkolu, tj. zjištění, zda každý žák rozumí zadání úlohy. Dále by si každý žák měl vypracovat geometrickou představu problému a také si uvědomit jeho řešení. Tedy nejprve by měl vyřešit problém geometricky pomocí rýsování.

V následující chvíli může žák zapnout počítač a začít pracovat na programování řešení problému.

Podrobněji:

- grafické vyznačení zadání úlohy, tj. grafické vyznačení zadaných objektů do obrázku micro:bitu;

- vyřešení geometrického problému pomocí rýsování a následné převedení řešení na LED diody micro:bitu;
- uvědomění si, které příkazy a funkce z programování budou k sepsání kódu potřeba.

5.7.5 Mezipředmětové vztahy

Vytvořené pracovní listy spojují oblasti programování a algoritmizace předmětu informatika s geometrií jako jednou oblastí matematiky.

Mezipředmětové vztahy mezi matematikou a informatikou také rozvíjí klíčové kompetence jako kompetence k řešení problémů a kompetence digitální.

5.7.6 Organizační doporučení

V ideálním případě by každý žák měl pracovat sám, lze ale vytvořit i skupinky po dvou nebo po třech žácích, v nichž by žáci úlohy řešili společně. Na druhou stranu by ale každý žák měl mít svůj pracovní list.

Pracovní list je vhodné tisknout ve formátu A4, aby bylo na pracovním listě vše dobře čitelné a aby měli žáci dostatek místa pro sepsání kódu.

Doporučený čas pro práci s jedním pracovním listem jsou dvě vyučovací hodiny. První hodina pro porozumění zadání a řešení úloh a druhá vyučovací hodina pro ověření správnosti programu a společnou diskuzi.

6 Propedeutika

6.1 GeoGebra

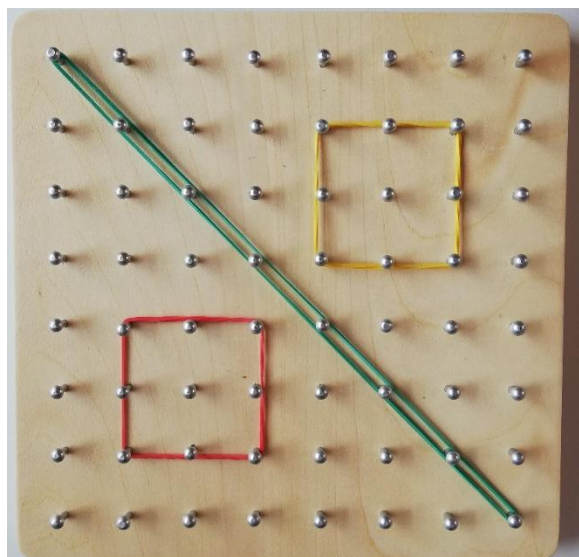
Propedeutikou pro řešení úloh z pracovních listů pomocí programování micro:bitu může být řešení úloh, pro které byly vhodně vytvořeny dynamické applety např. ve freewarovém softwaru GeoGebra.

V softwaru GeoGebra je možné vytvořit čtvercovou síť z izolovaných bodů, které odpovídají matici 5x5 LED diod micro:bitu. Do takto speciálně sestavené čtvercové sítě mohou žáci umísťovat geometrické objekty (bod, úsečku, přímku), ale i geometrické rovinné obrazce (trojúhelník, čtverec, obdélník apod.). Rovinné geometrické objekty i obrazce mohou být v GeoGebře předdefinovány a mohou být vloženy tak, aby s nimi bylo možné po nákresně pouze manipulovat (tj. posouvat je a otáčet s nimi). Proto úkoly zadávané žákům mohou spočívat např. v posouvání objektů v daném směru a o danou délku, či v otáčení objektů kolem daného středu např. na vyznačenou pozici, anebo v jejich zobrazení např. v osově souměrnosti s danou osou souměrnosti. Pokud jsou objekty/obrazce v softwaru GeoGebra předdefinované a žáci s nimi mohou po nákresně pouze manipulovat, zvládnou řešení obtížnostně odpovídajících úloh už i žáci 1. stupně základní školy.

6.2 Geoboard

Před použitím micro:bitu ve výuce informatiky na druhém stupni ZŠ, nebo na prvním stupni ZŠ se může např. při výuce některých témat z geometrie využít tzv. Geoboardu. Přitom Geoboardem zpravidla rozumíme čtvercovou desku s pravidelně rozmístěnými kolíky (tj. sousední kolíky jsou ve vodorovném i svislém směru od sebe stejně vzdáleny). Díky manuální činnosti s Geoboardem si žáci mohou vytvořit představu o rozmístění kolíků na desce Geoboardu, načež pro ně může být posléze orientace v matici LED diod micro:bitu mnohem snazší.

Geoboard může být také využit při vypracovávání úloh z výše popsanych, speciálně vytvořených pracovních listů i jako tzv. „deštivá varianta“, a to v momentě, kdy zradí technika (např. nepůjde elektrina, bude nefunkční připojení k internetu atd.). Pak pro řešení určitých částí úloh je možné využít Geoboardu, např. pro zobrazení zadání či následně i pro zobrazení řešení příslušné dané úlohy.



Obrázek 87: Ukázka zobrazení čtverce v osové souměrnosti s vyznačenou osou souměrnosti na Geoboardu

6.3 Geo:bit

Zajímavou aktivitou by pro žáky mohlo být i vymodelování speciálního Geoboardu v modelovacím softwaru a jeho následné vytištění na 3D tiskárně. Tuto aktivitu by však bylo možné uskutečnit pouze na školách, které vlastní 3D tiskárnu.

Žáci by nejprve ve vhodném programu pro tvorbu 3D modelů (např. ve freewaru Fusion 360) vytvořili návrh v podobě virtuálního 3D modelu, ten by poté přetransformovali do souboru s příponou *.gcode, který komunikuje s 3D tiskárnou, a následně by model vytiskli.

Vytvořený Geo:bit, jak jsme vymodelovaný a vytisknutý model, který je kombinací micro:bitu a Geoboardu, nazvaly, by se dal speciálně využívat v případě, kdy by nefungovala technika, nebo jako doprovodnou vizualizaci zadání problémů a jejich řešení.



Obrázek 88: Geo:bit

Závěr

Tato diplomová práce se zabývá tématem „Využití micro:bitů ve výuce matematiky“. Je uvedena představením základních poznatků vybraných témat z geometrie a zmínkou o programovacích jazycích, pomocí nichž lze programovat micro:bit. Dále následuje kapitola, v níž jsou popsány micro:bit a vývojové prostředí <https://makecode.microbit.org/>. V rámci této práce byly vytvořeny speciální pracovní listy, které vedou žáky k procvičování úloh z vybraných kapitol geometrie pomocí programování robota – micro:bitu, a dotazník, který následně zjišťuje zpětnou vazbu od žáků k vytvořeným pracovním listům a úlohám vložených v nich. Na základě získaných zpětných vazeb a podnětů od vyučujícího SŠ i od žáků, na kterých byly pracovní listy testovány, byly pracovní listy následně upraveny.

Teoretickými cíli této práce byly představení micro:bitu, oblastí programování a těch programovacích jazyků, pomocí nichž se micro:bit programuje. Dalším cílem teoretické části této diplomové práce bylo představení těch témat z geometrie, která jsou díky navrženým pracovním listům procvičována na základě tvorby programů pro micro-robota – micro:bit. Tyto cíle jsou naplněny v první, druhé a ve třetí kapitole této práce.

Cílem praktické části bylo vymyšlení vhodných úloh a sestavení z nich pracovních listů určených pro žáky základní a střední školy takovým způsobem, aby propojovaly oblasti matematiky a informatiky, konkrétně geometrie a programování. Pro žáky základní i střední školy byly navrženy vždy čtyři pracovní listy, tedy celkem bylo vytvořeno osm pracovních listů obsahujících úlohy k procvičování shodných zobrazení v rovině (osové a středové souměrnosti, posunutí, otočení) a analytické geometrie v rovině. Přitom pro žáky základní školy byly vytvořeny pracovní listy pouze s úlohami zaměřenými na procvičování shodných zobrazení v rovině. Pro žáky střední školy byly navrženy pracovní listy s tzv. nepolohovými úlohami, jejichž cílem bylo opět procvičování shodných zobrazení v rovině. Přitom pracovní list pro téma „Posunutí“ byl užit pro žáky obou stupňů škol a pro žáky střední školy byl navíc vytvořen pracovní list pro téma „Analytická geometrie v rovině“. Pro žáky obou stupňů škol byl ještě vytvořen pracovní list s pracovním názvem „Úvodní hodina“, jehož cílem bylo uvedení žáků do procvičované problematiky.

Dalšími cíli této práce bylo otestování pracovních listů při výuce a získání zpětné vazby od žáků, kteří pracovní listy vyplňovali. Na základě získané zpětné vazby pak pracovní listy smysluplně upravit.

Vypracování této diplomové práce mi bylo velkým přínosem, neboť jsem se seznámila s micro:bitem, který bych velmi ráda využívala i ve své budoucí praxi a také v rámci svého volného času. Ověřila jsem si, že mezi informatikou a matematikou je velmi tenká hranice, kterou díky úlohám podobným těm, které jsem zařadila do pracovních listů, mohou pocítit i žáci. Také věřím, že jsem našla způsob, jak žáky motivovat a zaujmout při hodinách matematiky nebo informatiky. Myslím, že názorná ukázka propojení předmětů, respektive některých jejich oblastí žákům umožňuje na předměty nahlédnout více komplexněji a nepohlížet na ně pouze jako na separované části vzdělávání.

V rámci této práce jsem se naučila vytvořit dotazník pro získání zpětné vazby. Tyto znalosti nemusím využít pouze k získání zpětné vazby pro vytvořené pracovní listy, ale mohu je využít také pro získání jakékoli zpětné vazby od žáků. Krátkým dotazníkovým šetřením mohu od žáků obdržet cenné informace, které mohou zlepšit společný vzdělávací proces.

Při vyhodnocování odpovědí z dotazníku jsem si uvědomila důležitost jednoznačnosti kladených otázek a formální stránky pro jeho komfortní vyplnění žáky.

Přestože dotazník prošel pilotním šetřením, které mělo zjistit jeho nedostatky, i jeho ostrá verze vykazovala jistý prostor pro zlepšení. Např. se ukázalo účelné vytvořit dotazník pro každý pracovní list zvlášť.

Na druhou stranu si myslím, že byl dotazník dobře sestaven, protože jeho výsledky mi daly jasný a srozumitelný obraz o použitelnosti pracovních listů.

Během zpracovávání dat mě některé odpovědi velmi pobavily. Např. na otázku: „Uveď důvod, proč se ti uvedená úloha jevila jako obtížná/těžká.“ zazněla odpověď „Protože jsem až moc inteligentní.“

Z některých odpovědí jsem měla velmi pozitivní pocit. Např. na otázku „Co bys změnil/a na pracovním listu?“ byla uvedena odpověď „Nic, bylo to zajímavé. Aspoň nějaká výzva.“

Závěrem bych chtěla poznamenat, že micro:bit se dá využít v mnoha školních předmětech, ale i v mnoha dalších oblastech matematiky, např. v kombinatorice nebo logice. Tvorbou speciálních pracovních listů a propojováním oblastí informatiky a matematiky se plánuji věnovat i ve svém budoucím učitelském působení.

Zdroje

Literatura:

- [1] ODVÁRKO, O. – KADLEČEK, J.: *Přehled matematiky pro základní školy a víceletá gymnázia*. PROMETHEUS, Praha 2004. 270 str. ISBN: 80-7196-276-7
- [2] VONDRA, J.: *Matematika pro SŠ – 7.díl A; Analytická geometrie, Učebnice*. Didaktis, Brno 2016. 88 str. ISBN: 978-80-7358-236-4
- [3] GAVORA, P.: *Úvod do pedagogického výzkumu*. Paido, Brno 2000. 207 str. ISBN: 80-85931-79-6
- [4] KOČANDRLE, M. – BOČEK, L.: *Matematika pro gymnázia – Analytická geometrie*. PROMETHEUS, Praha 2012. 220 str. ISBN: 978-80-7196-390-5
- [5] PECH, J. – PRŠALA, J. – VANÍČEK J. – NOVÁK, M.: *Robotika pro základní školy: programujeme micro:bit pomocí Makecode*. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta 2021. 79 str. ISBN: 978-80-7394-851-1. Dostupné na: <https://imysleni.cz/ucebnice/18-robotika-pro-zakladni-skoly-programujeme-micro-bit-pomoci-makecode>
- [6] PECINOVSKÝ, R.: *Python - Kompletní příručka jazyka pro verzi 3.9*. Grada, Praha 2020. 480 str. ISBN: 978-80-271-1269-2
- [7] PÍSEK, S.: *JavaScript - Efektní nástroj oživení WWW stránek*. Grada, Praha 2001. 231 str. ISBN: 80-247-0014-X
- [8] SLOUKA, J.: *Geometrie pro 5. - 9. ročník ZŠ a víceletá gymnázia*. FIN, Olomouc 1993. 192 str. ISBN: 80-85572-53-2
- [9] PALKOVÁ, M. – ZEMEK, V.: *Průvodce matematikou 1 aneb co byste měli znát z numerické matematiky ze základní školy*. Didaktis, Brno 2009. 200 str. ISBN 978-80-7358-274-6
- [10] PALKOVÁ, M. a kol.: *Průvodce matematikou 2 aneb co byste měli znát z geometrie ze základní školy*. Didaktis, Brno 2007. 135 str. ISBN 978-80-7358-275-3
- [11] PAVLAS, J.: *Metodický list jako didaktický prostředek*. Trendy ve vzdělávání, Olomouc 2008. str. 205–208. Dostupné na: <https://tvv-journal.upol.cz/pdfs/tvv/2008/01/49.pdf>
- [12] Rámcový vzdělávací program pro základní vzdělávání [online]. Praha: MŠMT, 2021.

Dostupné na: <https://www.edu.cz/rvp-ramcove-vzdelavaci-programy/ramcove-vzdelavaci-program-pro-zakladni-vzdelavani-rvp-zv/>

- [13] Rámcový vzdělávací program pro obor vzdělání 18-20-M/01. Informační technologie [online]. Praha: MŠMT.

Dostupné na: <http://rvp.cz/informace/wp-content/uploads/2009/09/RVP-7241M01.pdf>

- [14] PECH, J. – NOVÁK, M.: *Robotika pro střední školy: Programujeme micro:bit pomocí Pythonu*. Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, České Budějovice 2020. 40 str. ISBN 978-80-7394-787-3

Internetové zdroje:

- [I] BBC, Make It Digital. *What is the BBC micro:bit?* [Online] Dostupné na: <https://www.bbc.co.uk/programmes/articles/4hVG2Br1W1LKCmw8nSm9WnQ/the-bbc-micro-bit>
- [II] BBC, Make It Digital. [Online] Dostupné na: <https://www.bbc.co.uk/programmes/p02wl84w>
- [III] Blockly, *Introduction to Blockly*. [online] Dostupné na: <https://developers.google.com/blockly/guides/overview>
- [IV] 14. Programovací jazyky. 2022. Dostupné na: <http://www.ivt.mzf.cz/seminar/14-programovaci-jazyky/>
- [V] IT SLOVNÍK.CZ: *Co je to Proměnná?* 2022. Dostupné na: <https://it-slovník.cz/pojem/promenna>
- [VI] W3Schools JavaScript. [Online] Dostupné na: <https://www.w3schools.com/js>
- [VII] W3Schools Python. [Online] Dostupné na: <https://www.w3schools.com/python>

Obrázky:

Obrázek 87: Ukázka Geoboardu – Autor: Mgr. Daniela Bímová, Ph.D.

Obrázek 88: Geo:bit - Autor: Mgr. Daniela Bímová, Ph.D.

Přílohy

Pracovní listy

Dotazník

Micro:bit

<https://makecode.microbit.org/>

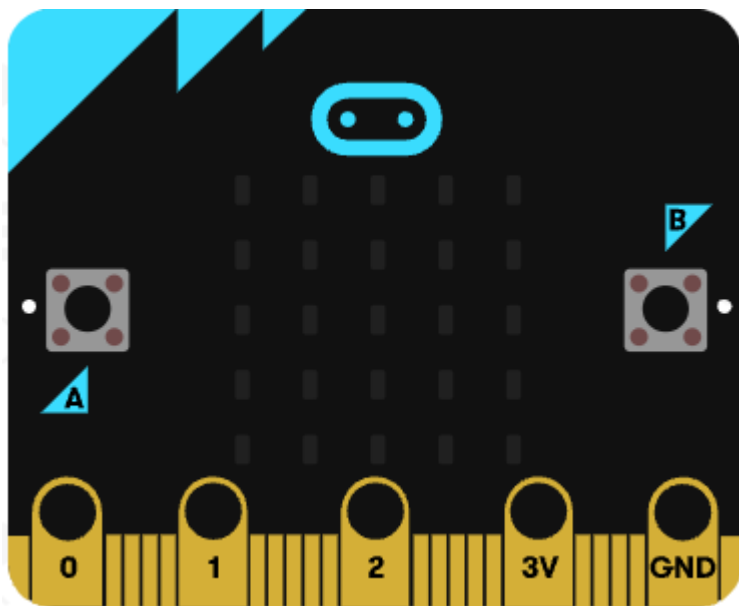
1. úloha

Jak funguje číslování řádků a sloupců?

2. úloha

Rozsviť bod na souřadnicích [1, 3]

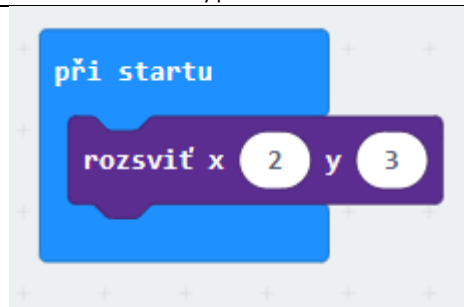
Označ si, která dioda se rozsvítí?



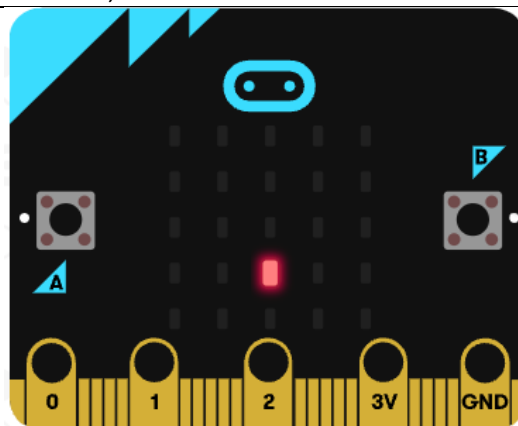
Vzorové řešení

Rozsviť bod na souřadnicích [2, 3]

Jak vypadá kód:



Znázornění, kde se bod v matici LED diod zobrazí:

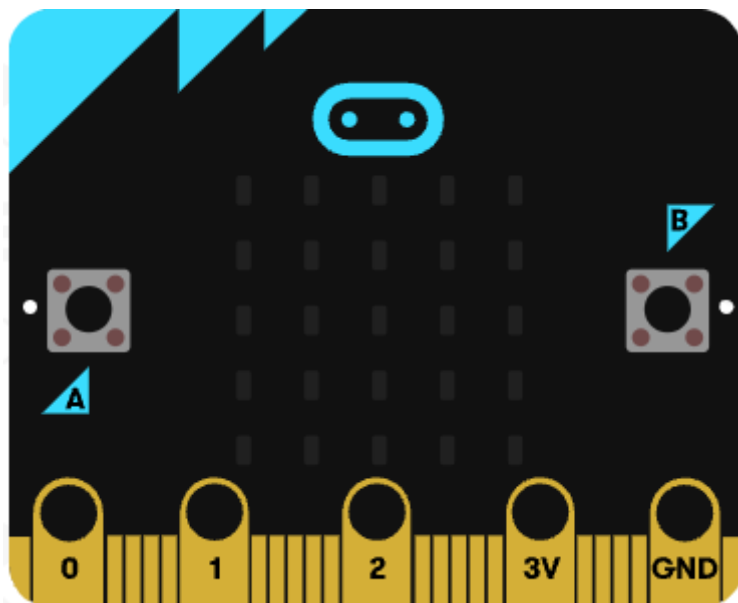


Existuje více možností, jak rozsvítit bod. Dokážeš najít nějaký další způsob?

3. úloha

Rozsviť bod na souřadnicích [3, 4] s jasem 150.

Označ si, která dioda se rozsvítí?

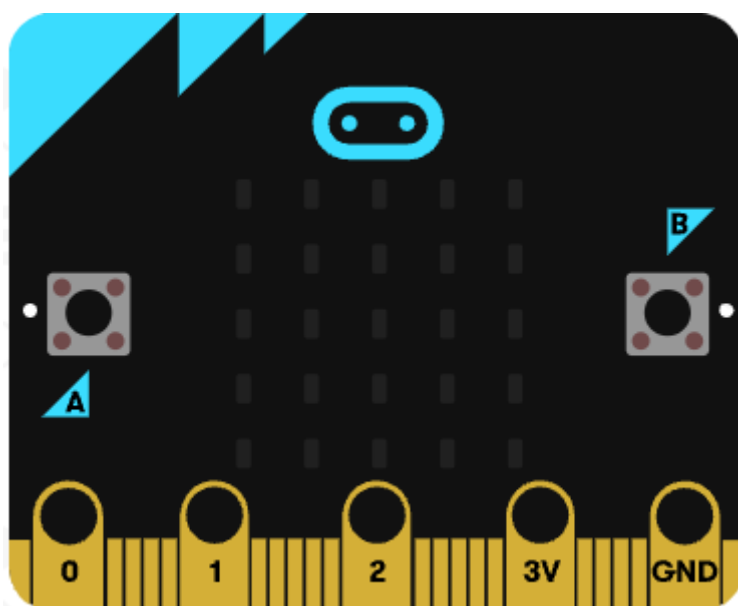


4. úloha

Po stisknutí tlačítka **A** rozsviť obvod čtverce se stranou 3 jednotky* libovolně umístěný na micro:bit

** jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami*

Označ si, která dioda se rozsvítí?

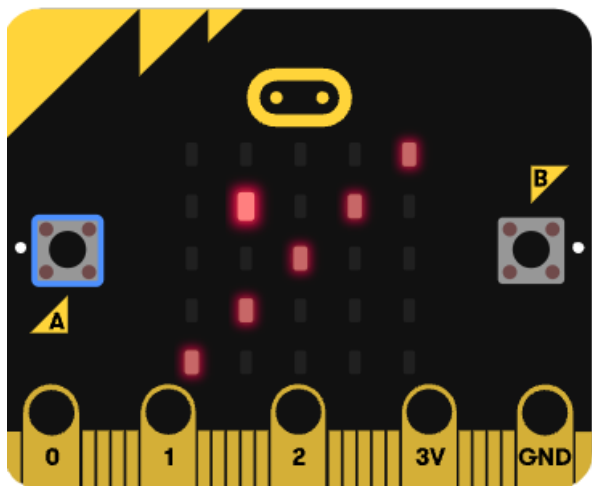


Micro:bit - Osová souměrnost

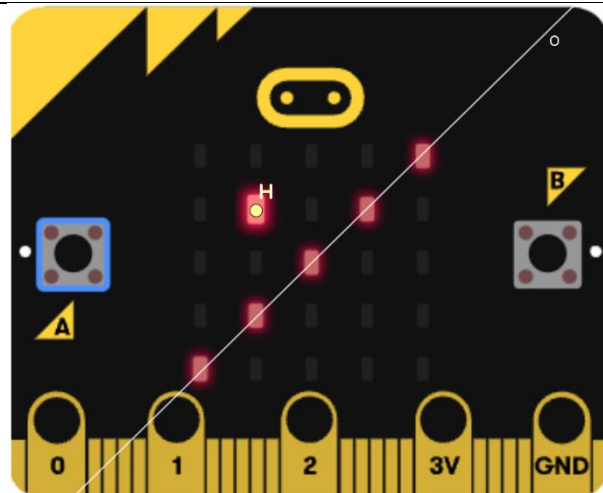
1. úloha

Kliknutím na tlačítko **A** vybereš zadání pro zobrazení bodu H v dané osově souměrnosti. Najdi a rozsviř bod H' , který je osově souměrný s bodem H . Osa o souměrnosti je vyznačena slaběji.

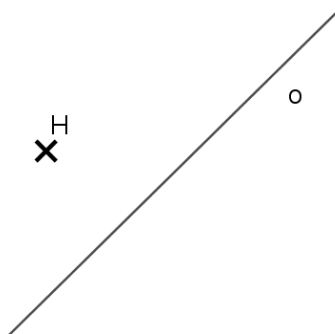
Micro:bit



Geometrické objekty v micro:bitu



Převedení problému na geometrický



Napište souřadnice

Doplň souřadnice do kódu tak, aby se po kliknutí na tlačítko **B** zobrazil obraz H' bodu H .

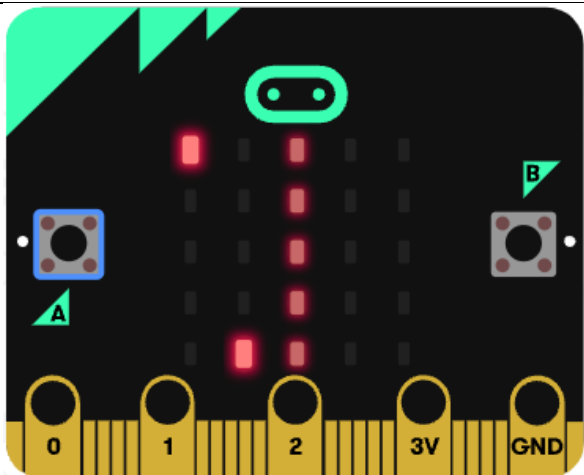


Micro:bit - Osová souměrnost

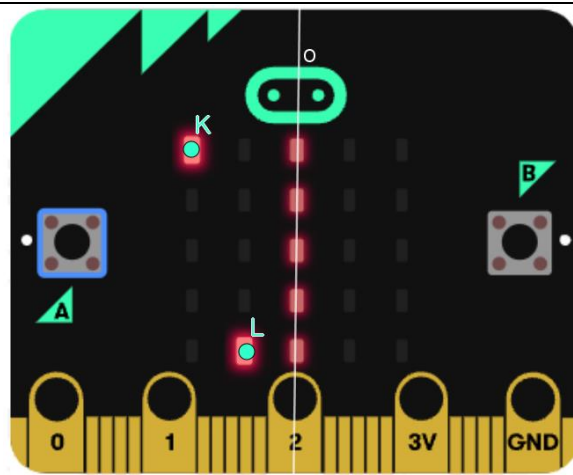
2. úloha

Po druhém kliknutí na tlačítko **A** vybereš zadání pro zobrazení úsečky KL v osově souměrnosti. Zobrazení úsečky KL v osově souměrnosti. Osa o je vyznačena slaběji.

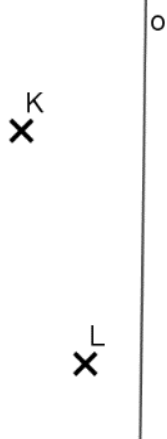
Micro:bit



Geometrické objekty v micro:bitu

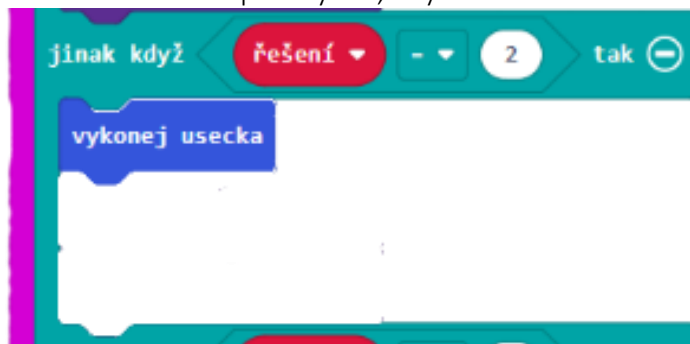


Převedení problému na geometrický



Napište, na jakých souřadnicích se rozsvítí obrazy krajních bodů dané úsečky.

Doplň do kódu blokové příkazy tak, aby se rozsvítilo řešení úlohy.

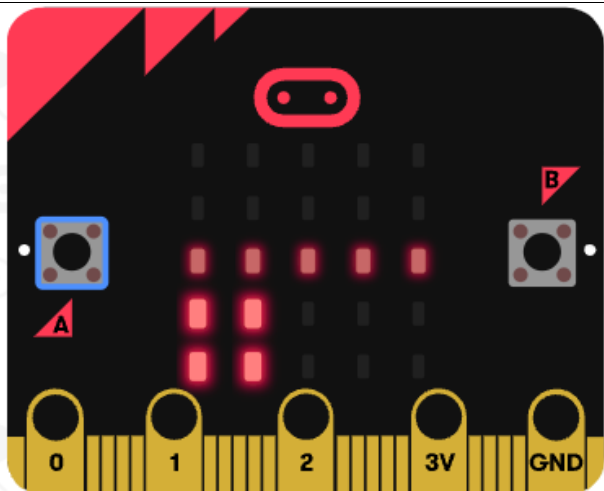


Micro:bit - Osová souměrnost

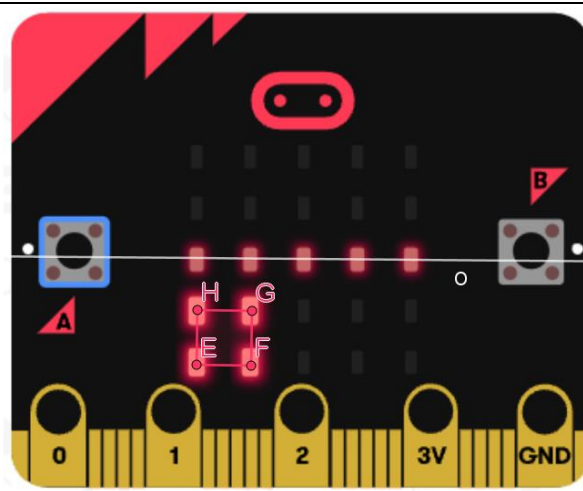
3. úloha

Po třetím kliknutí na tlačítko **A** vybereš zadání pro zobrazení čtverce *EFGH* pomocí osově souměrnosti. Zobraz čtverec *EFGH* v osově souměrnosti. Osa *o* je vyznačena slaběji.

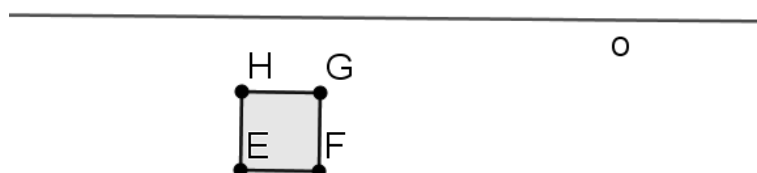
Micro:bit



Geometrické objekty v micro:bitu

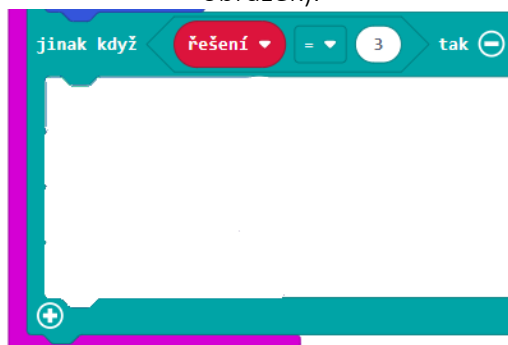


Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí obrazy vrcholů daného čtverce.

Vytvoř kód tak, aby se rozsvítilo řešení i zadání úlohy. Kód dodělej do větve kódu, kde je „řešení = 3“ (viz obrázek).

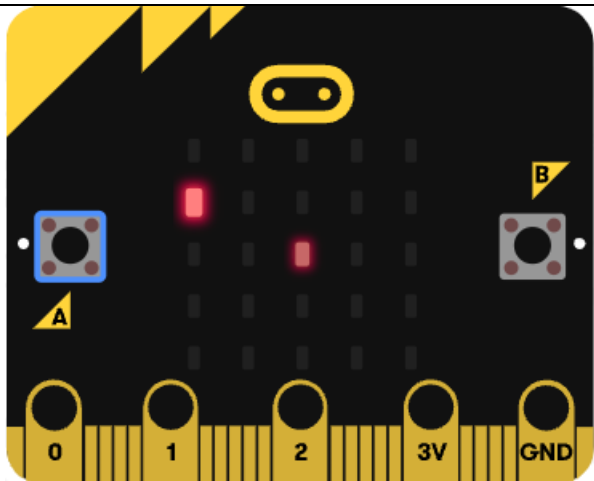


Micro:bit - Středová souměrnost

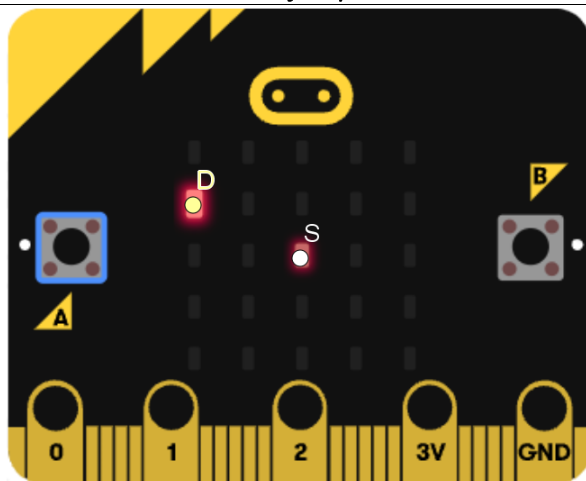
1. úloha

Kliknutím na tlačítko **A** vybereš zadání pro zobrazení bodu ve středové souměrnosti. Najdi bod D' , který je středově souměrný s bodem D . Střed S je vyznačen slaběji.

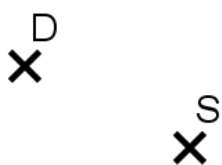
Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí středově souměrný bod.

Doplň souřadnice do kódu tak, aby se po kliknutí na tlačítko **B** zobrazil středově souměrný bod D' .

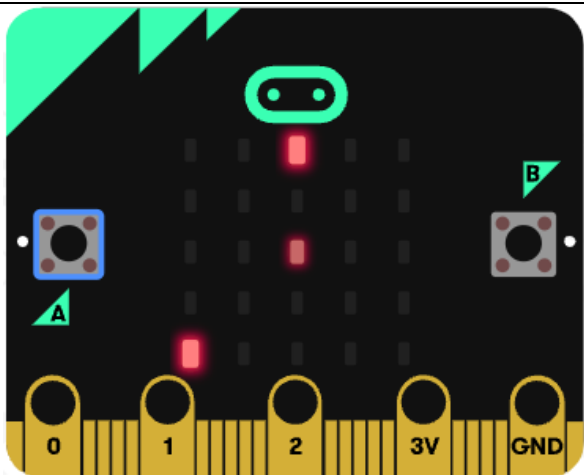


Micro:bit - Středová souměrnost

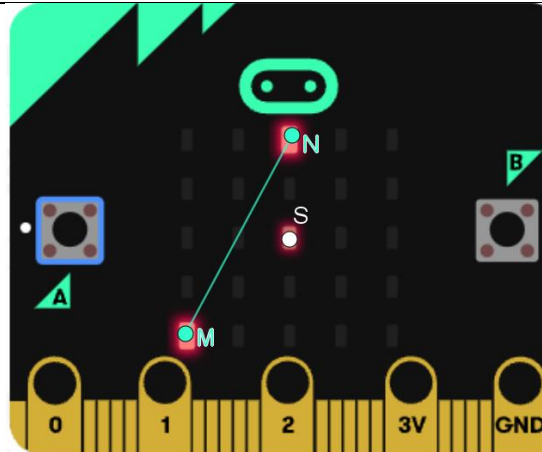
2. úloha

Po druhém kliknutí na tlačítko **A** vybereš zadání pro zobrazení úsečky MN v dané středové souměrnosti. Zobraz úsečku MN ve středové souměrnosti. Střed S je vyznačen slaběji.

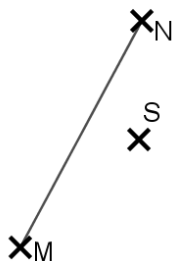
Micro:bit:



Geometrické objekty v micro:bitu

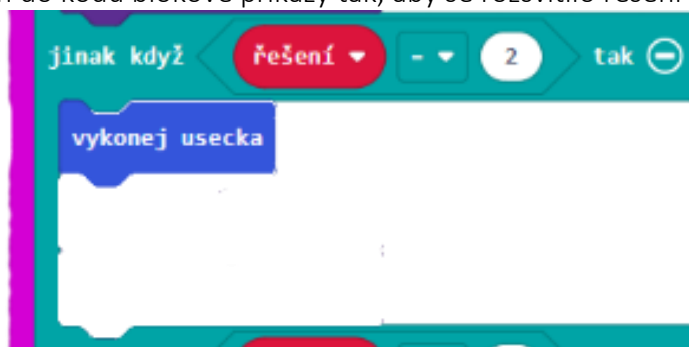


Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí obrazy krajních bodů dané úsečky?

Doplň do kódu blokové příkazy tak, aby se rozsvítilo řešení úlohy.

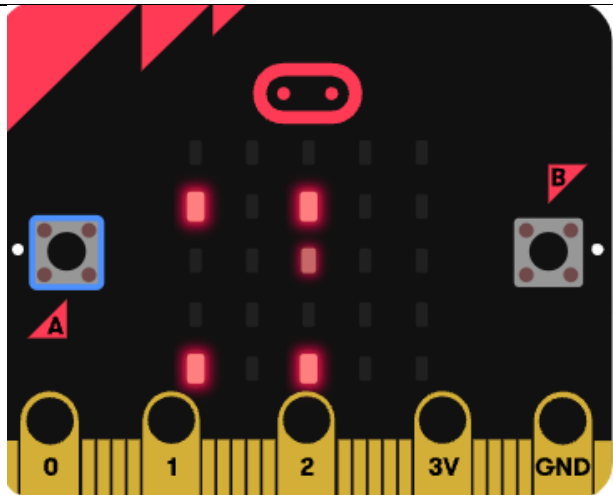


Micro:bit - Středová souměrnost

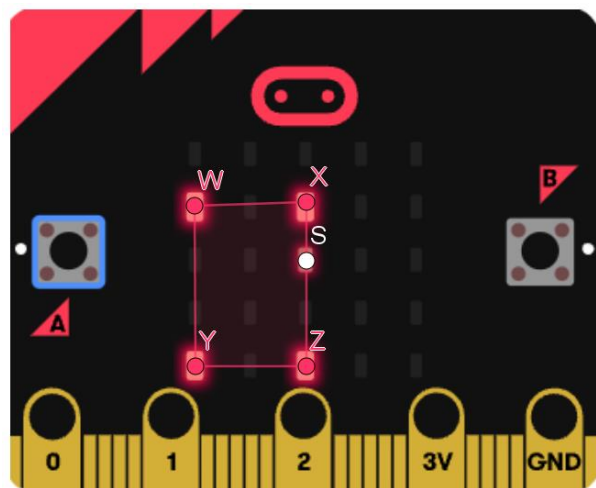
3. úloha

Po třetím kliknutí na tlačítko **A** vybereš zadání pro zobrazení obdélníka WYXZ v dané středové souměrnosti. Zobraz obdélník WYXZ ve středové souměrnosti. Střed *S* je vyznačen slaběji.

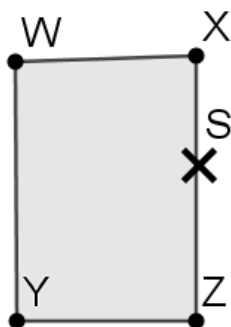
Micro:bit:



Geometrické objekty v micro:bitu

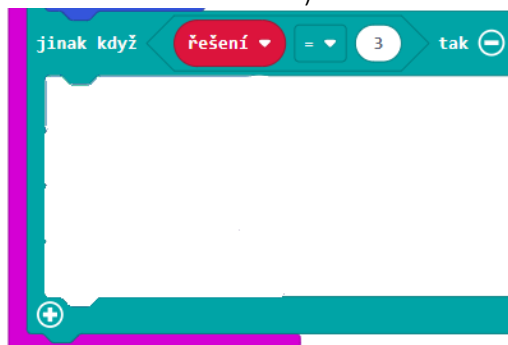


Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí přenesené body daného obdélníka?

Vytvoř kód tak, aby se rozsvítilo řešení i zadání úlohy. Kód dodělej do větve kódu, kde je „řešení = 3“ (viz obrázek).

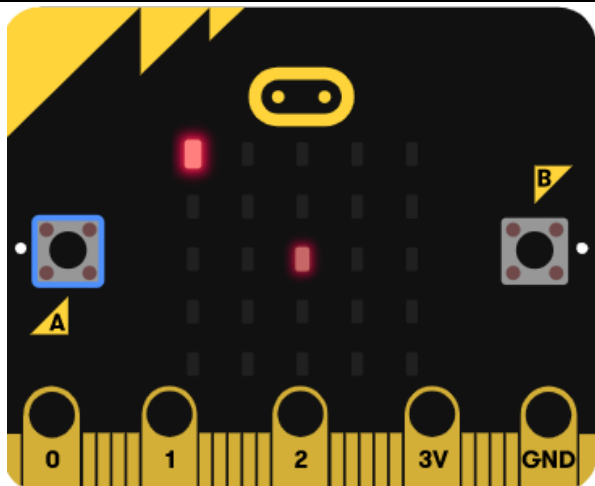


Micro:bit - Otočení

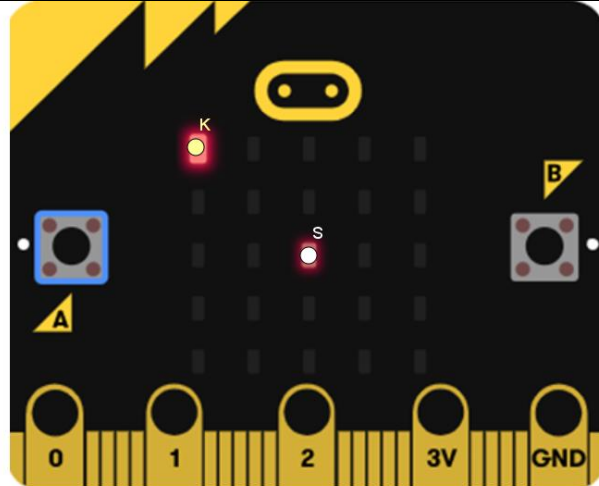
1. úloha

Kliknutím na tlačítko **A** vybereš zadání pro otočení bodu. Otoč tento bod o $+90^\circ$. Střed otočení **S** je rozsvícen slaběji.

Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:

$\times^K$

$\times^S$

Napište, na jakých souřadnicích se rozsvítí otočený bod.

Doplň souřadnice do kódu tak, aby se po kliknutí na tlačítko **B** zobrazil otočený bod.

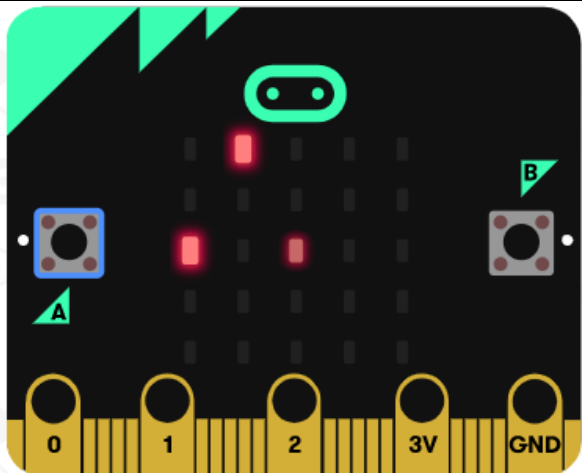


Micro:bit - Otočení

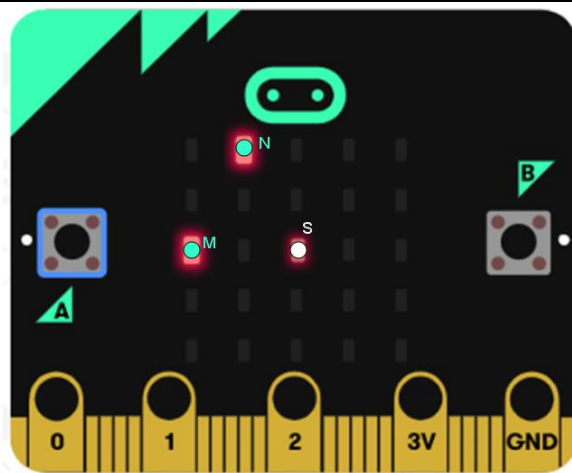
2. úloha

Po druhém kliknutí na tlačítko **A** se aktivuje zadání pro otočení úsečky MN . Otoč tuto úsečku o -90° . Střed otočení je rozsvícen slaběji.

Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:

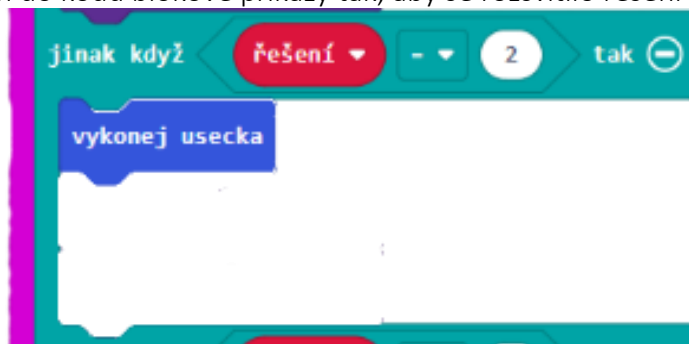
x^N

x^M

x^S

Napište, na jakých souřadnicích se rozsvítí krajní body otočené úsečky.

Doplň do kódu blokové příkazy tak, aby se rozsvítilo řešení úlohy.

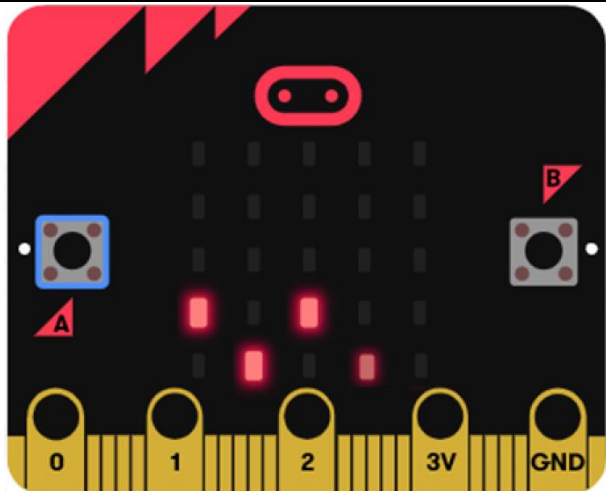


Micro:bit - Otočení

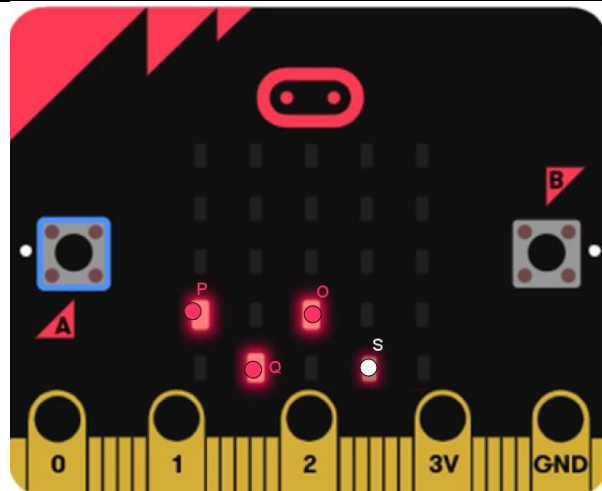
3. úloha

Po třetím kliknutí na tlačítko **A** se zobrazí zadání pro otočení trojúhelníku *OPQ*. Otoč tento trojúhelník o $+90^\circ$. Střed otočení je rozsvícen slaběji.

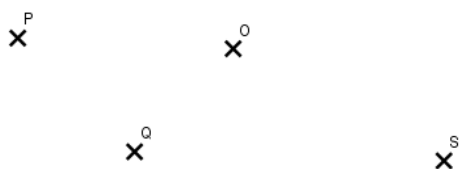
Micro:bit:



Geometrické objekty v micro:bitu

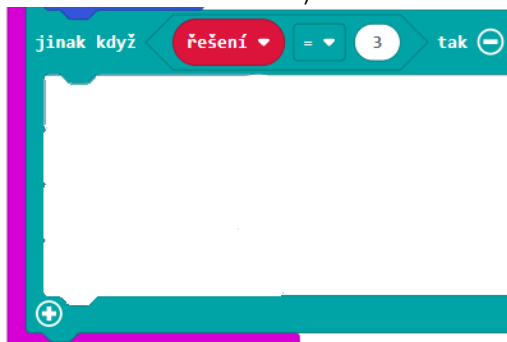


Převedení problému na geometrický:



Napište, na jakých souřadnicích se rozsvítí otočené vrcholy otočeného trojúhelníku.

Vytvoř kód tak, aby se rozsvítlo řešení i zadání úlohy. Kód dodělej do větve kódu, kde je „řešení = 3“ (viz obrázek).



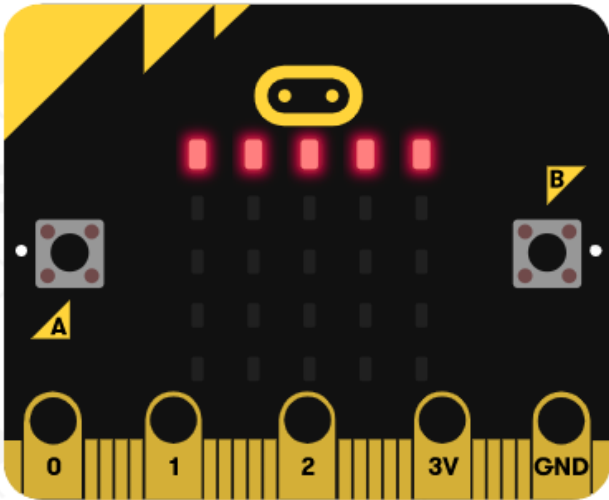
Micro:bit - Posunutí

1. úloha

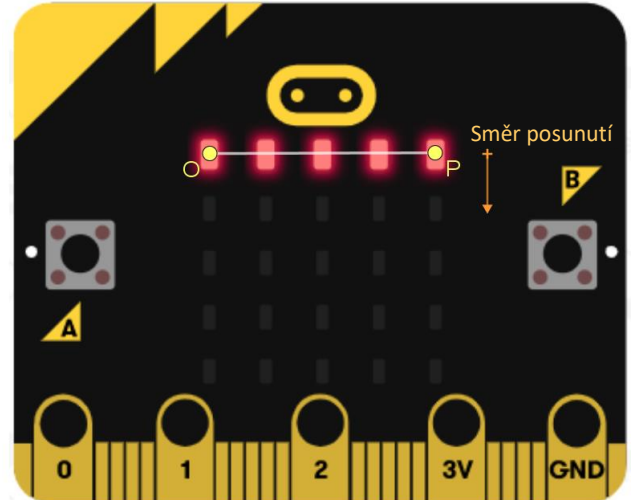
Zobraz vodorovnou úsečku OP (viz obrázek). Pro stisknutí tlačítka **A** vytvoř program, který bude každé 3 vteřiny posouvat úsečku OP vždy o jednu jednotku* ve svislém / vertikálním směru dolů.

**jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru*

Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:



Jak bude vypadat výsledný kód, pokud chceme použít proměnné?

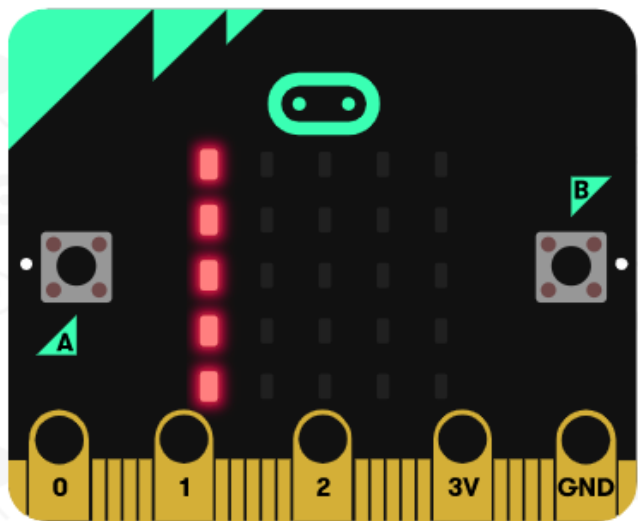
Micro:bit - Posunutí

2. úloha

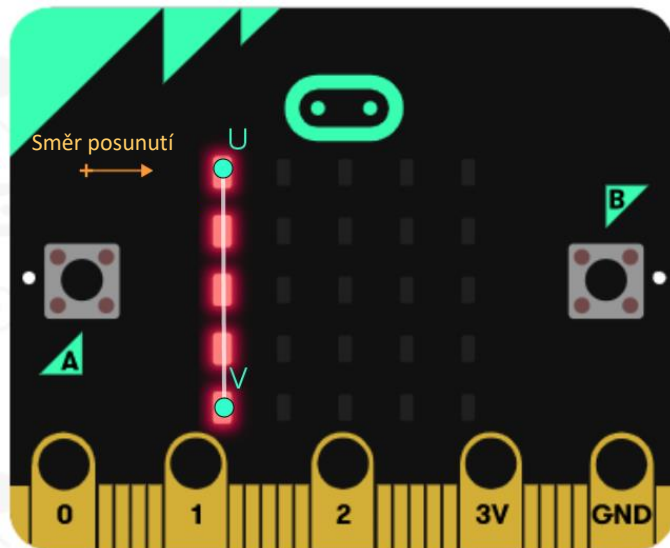
Zobraz svislou úsečku UV (viz obrázek). Pro stisknutí tlačítka **B** vytvoř program, který bude každé 3 vteřiny posouvat úsečku UV vždy o jednu jednotku* ve vodorovném směru doprava.

**jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru*

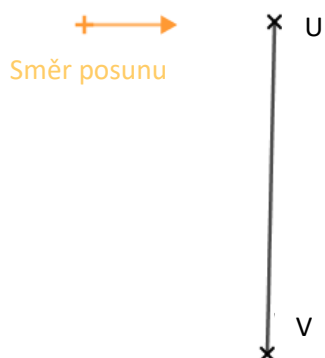
Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:



Jak bude vypadat výsledný kód, pokud chceme použít proměnné?

Micro:bit - Posunutí

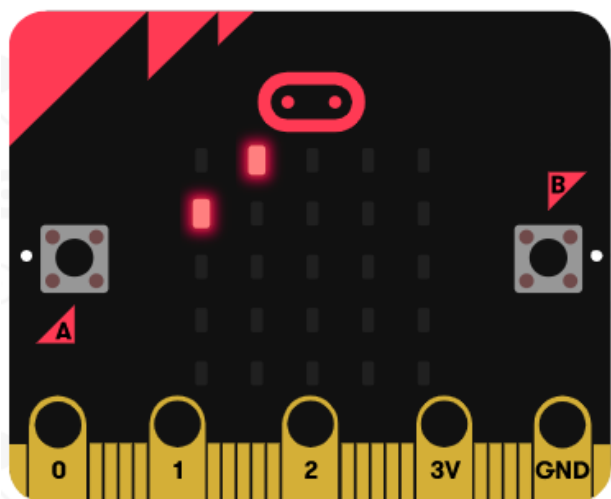
3. úloha

Zobraz šikmou úsečkou KL (viz obrázek). Pro stisknutí obou tlačítek **A**, **B** vytvoř program, který bude každé 3 vteřiny posouvat úsečku KL vždy o jednu jednotku* v diagonálním (úhlopříčném) směru doprava dolů.

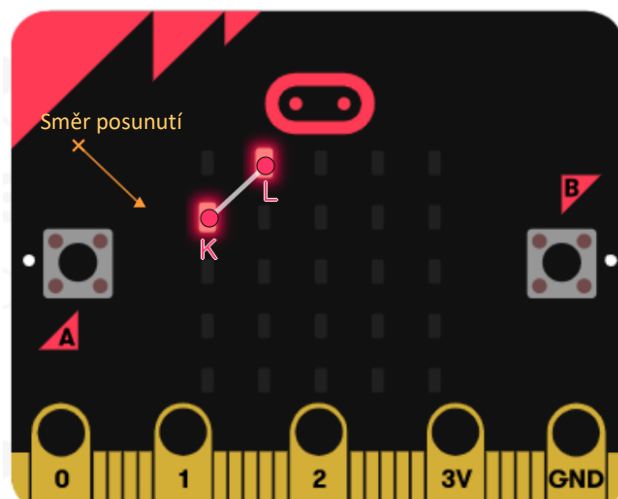
(Posunutí v diagonálním směru znamená složení posunutí o jednu jednotku ve vodorovném směru a o jednu jednotku ve svislém směru.)

* *jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru*

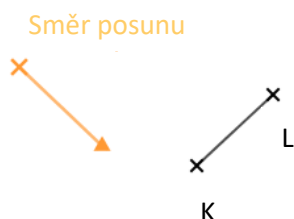
Micro:bit:



Geometrické objekty v micro:bitu



Převedení problému na geometrický:



Jak bude vypadat výsledný kód, pokud chceme použít proměnné?

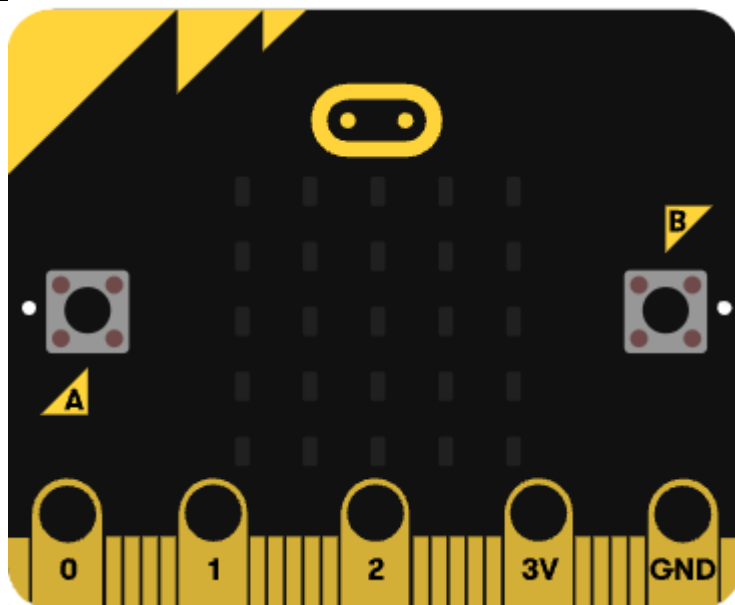
Micro:bit - Osová souměrnost

1. úloha

Vhodně umísti (rozsviť) svislou osu o tak, aby bylo možné zobrazit v osově souměrnosti dané osou o libovolně umístěný čtverec $ABCD$ se stranou délky 1 jednotka* na čtverec $A'B'C'D'$. Přitom čtverec $ABCD$ umístěte tak, aby neměl žádný bod společný s osou o a aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy čtverce $A'B'C'D'$. Osu o rozsviť s nižším jasnem než vrcholy obou čtverců.

** jednou jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru*

Micro:bit pro vyznačení rozsvícené osy o a vrcholů čtverců $ABCD$ a $A'B'C'D'$



Kód v JavaScript

Jak vypadá kód v JavaScript?

Zapiš si, jak vypadají jednotlivé příkazy:

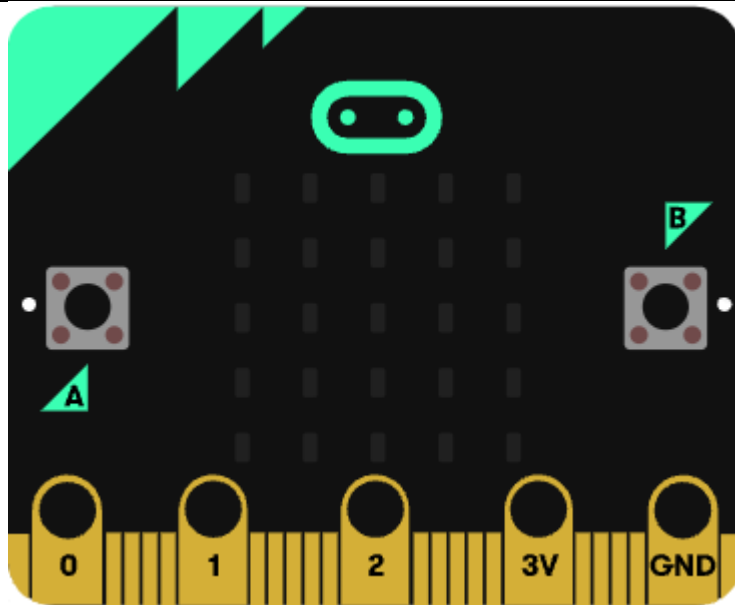
Micro:bit - Osová soměrnost

2. úloha

Vhodně umístí (rozsvítí) osu o tak, aby se v osově souměrnosti dané osou o dal zobrazit libovolně umístěný obecný trojúhelník KLM , který má s osou o společný právě jeden bod. Přitom trojúhelník KLM umístí tak, aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy jeho obrazu $K'L'M'$. Osu o rozsvítí s nižším jasnem než vrcholy obou trojúhelníků.

Vytvoř cyklus, který třikrát rozsvítí a zhasne trojúhelník KLM i jeho obraz $K'L'M'$.

Micro:bit pro vyznačení rozsvícené osy o a vrcholů trojúhelníků KLM a $K'L'M'$



Kód v JavaScript

Jak vypadá kód v JavaScript?

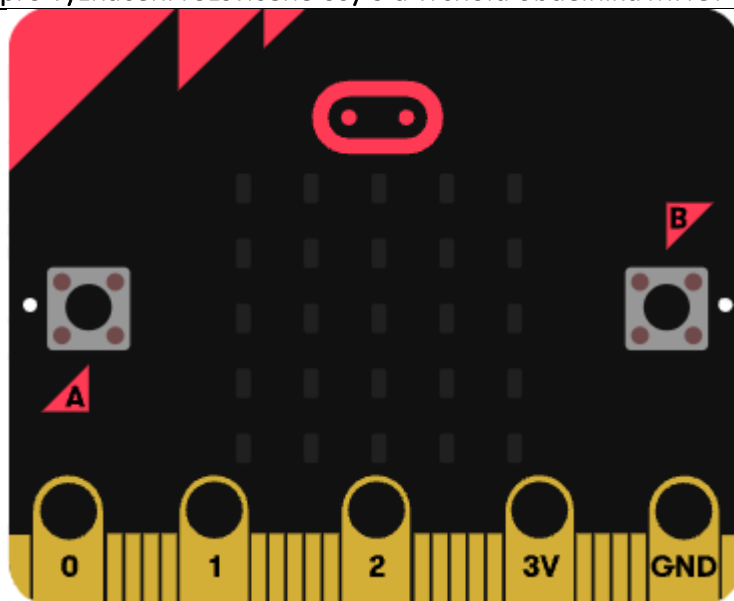
Micro:bit - Osová souměrnost

3. úloha

Vhodně umístí (rozsvítí) osu o tak, aby se v osově souměrnosti dané osou o dal zobrazit libovolně umístěný obdélník $MNOP$, který má s osou o společné právě dva body. Přitom obdélník $MNOP$ umístí tak, aby byly rozsvíceny všechny jeho vrcholy i všechny vrcholy jeho obrazu $M'N'O'P'$. Osu o rozsvítí s nižším jasnem než vrcholy obou obdélníků.

Postupně rozsvítí a zhasní diody na obvodu celého obdélníku tak, aby vždy svítila právě jedna dioda. Dále rozsvítí a zhasní sově souměrnou diodu k právě rozsvícené vzorové diodě.

Micro:bit pro vyznačení rozsvícené osy o a vrcholů obdélníků $MNOP$ a $M'N'O'P'$



Kód v JavaScript

Jak vypadá kód v JavaScript?

Micro:bit - Středová souměrnost

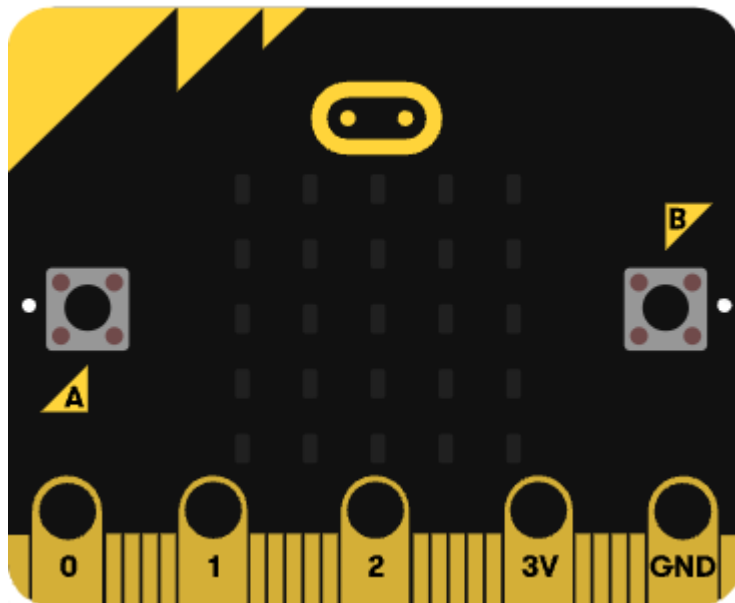
1. úloha

Vhodně umístí (rozsvítí) střed souměrnosti S tak, aby se ve středové souměrnosti dané středem S dal zobrazit libovolně umístěný čtverec $ABCD$ se stranou délky 1 jednotka* na čtverec $A'B'C'D'$. Přitom střed S zvol tak, aby nebyl bodem čtverce $ABCD$ a aby byly rozsvíceny všechny vrcholy čtverce $ABCD$ a jeho obrazu $A'B'C'D'$. Střed S rozsvít s nižším jasnem než vrcholy obou čtverců.

Dále vytvoř kód, který po stisknutí tlačítka **A** změní velikost strany čtverce o jednu jednotku, a to postupně dvakrát. Po třetím stisknutí tlačítka **A** se opět zobrazí čtverec se stranou 1 jednotka.

* jednu jednotkou rozumíme vzdálenost mezi dvěma sousedními diodami v horizontálním či ve vertikálním směru

Micro:bit pro vyznačení vysvíceného středu S a vrcholů čtverců $ABCD$ a $A'B'C'D'$



Kód v JavaScript

Jak vypadá kód v JavaScript?

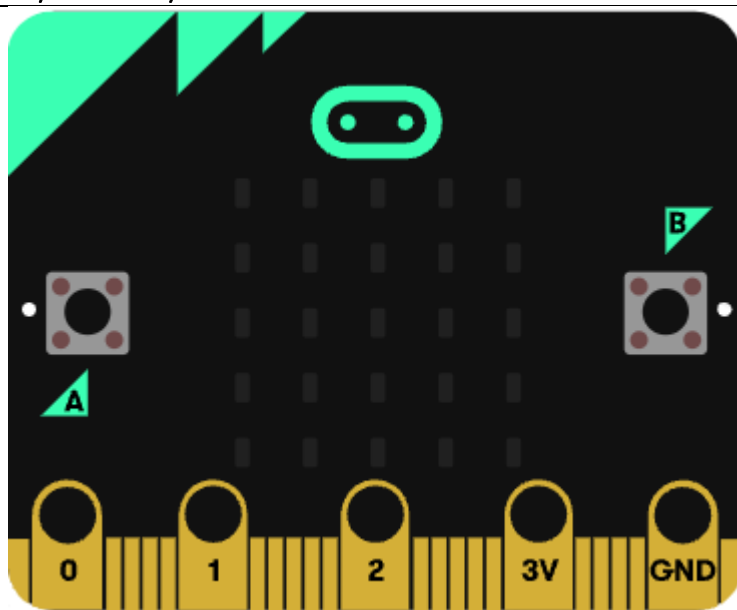
Micro:bit - Středová souměrnost

2. úloha

Vhodně umístí (rozsvítí) střed souměrnosti S tak, aby se ve středové souměrnosti dané středem S dal zobrazit libovolně umístěný obdélník $MNOP$. Přitom zvol střed S tak, aby nebyl bodem obdélníku $MNOP$ a aby byly rozsvíceny všechny vrcholy obdélníku $MNOP$ i jeho obrazu $M'N'O'P'$. Střed S rozsvítí s nižším jasnem než vrcholy obou obdélníků.

Dále vytvoř kód, který bude postupně měnit pozici obdélníku tak, aby se rozměry obdélníku neměnily.

Micro:bit pro vyznačení vysvíceného středu S a vrcholů obdélníků $MNOP$ a $M'N'O'P'$



Kód v JavaScript

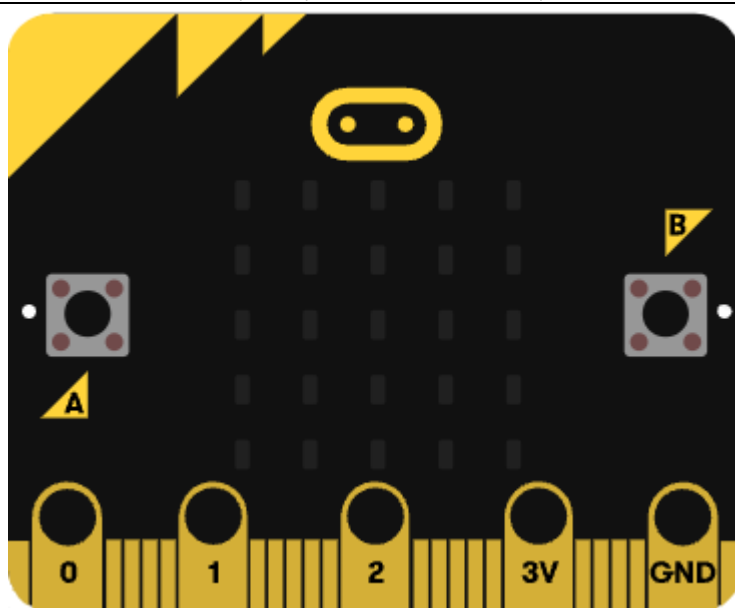
Jak vypadá kód v JavaScript?

Micro:bit - Otočení

1. úloha

Umístí střed otočení *S* na souřadnice [2, 2] micro:bitu. Střed *S* rozsviť s nižším jasnem. Dále rozsviť vrcholy libovolného trojúhelníku *ABC*. Vytvoř kód tak, aby se kliknutím na tlačítko **A** otočil trojúhelník *ABC* kolem středu *S* o orientovaný úhel $+90^\circ$ a kliknutím na tlačítko **B** trojúhelník *ABC* otočil o orientovaný úhel -90° . Přitom umístěte trojúhelník *ABC* na matici LED diod tak, aby všechny jeho vrcholy i vrcholy jeho obrazů byly rozsvíceny.

Micro:bit pro vyznačení rozsvícených diod



Kód v JavaScript

Jak vypadá kód v JavaScript?

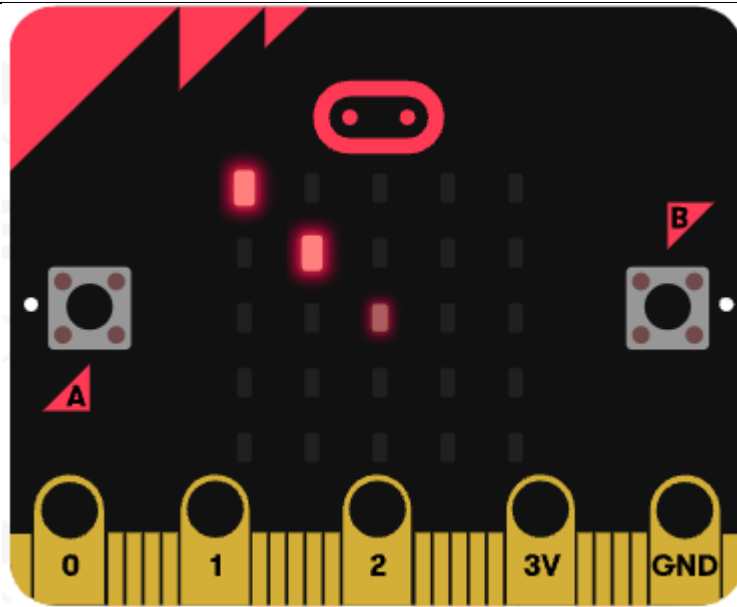
Micro:bit - Otočení

2. úloha

Umístí střed otočení S na souřadnice $[2, 2]$ micro:bitu. Střed S rozsviť s nižším jasnem. Dále s vyšším jasnem rozsviť dvě diody na souřadnicích $[0, 0]$, $[1, 1]$ micro:bitu a vytvoř cyklus, který bude postupně rozsvěcet a zhasínat dvě dané diody otáčející se kolem středu S ve zvoleném směru (tj. buď po směru nebo proti směru hodinových ručiček).

O kolik úhlových stupňů se v každém kroku každá z rozsvícených diod otočí?

Micro:bit pro vyznačení rozsvícených diod



Kód v JavaScript

Jak vypadá kód v JavaScript?

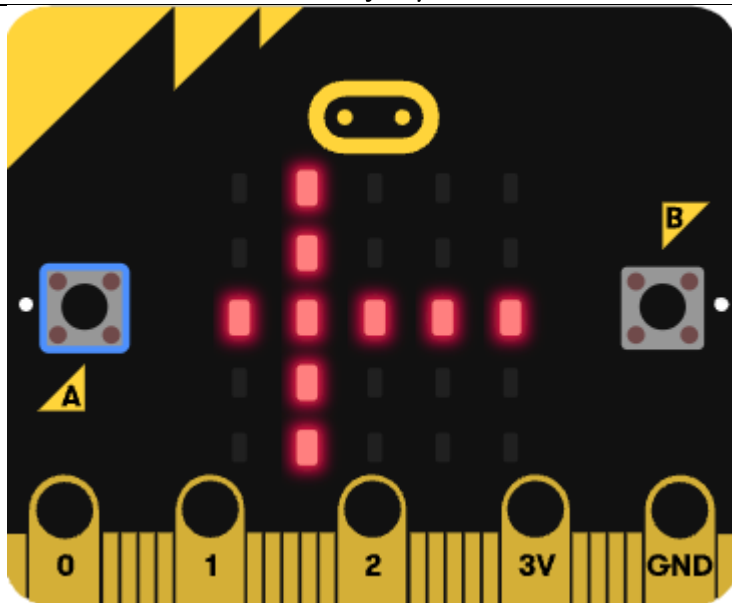
Micro:bit - Analytická geometrie

1. úloha

Kliknutím na tlačítko **A** se zobrazí souřadnicový systém.

- Uprav kód tak, aby se osa x a osa y zobrazily s jasem 155.
- Rozsviť body K a L , kde $K = [2, 2]$ a $L = [1, -2]$
- Napiš obecnou rovnici přímky procházející body K a L .

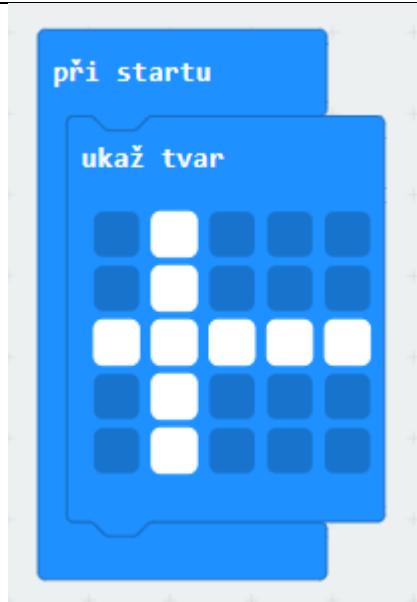
Geometrické objekty v micro:bitu



Kód v makecode.microbit.com

Kód pomocí JavaScript

Jak vypadá tento blokový kód v JavaScript?



Odvození obecné rovnice přímky:

Micro:bit - Analytická geometrie

2. úloha

Umístí vhodně souřadnicový systém, který se zobrazí po kliknutí na tlačítko **B**.

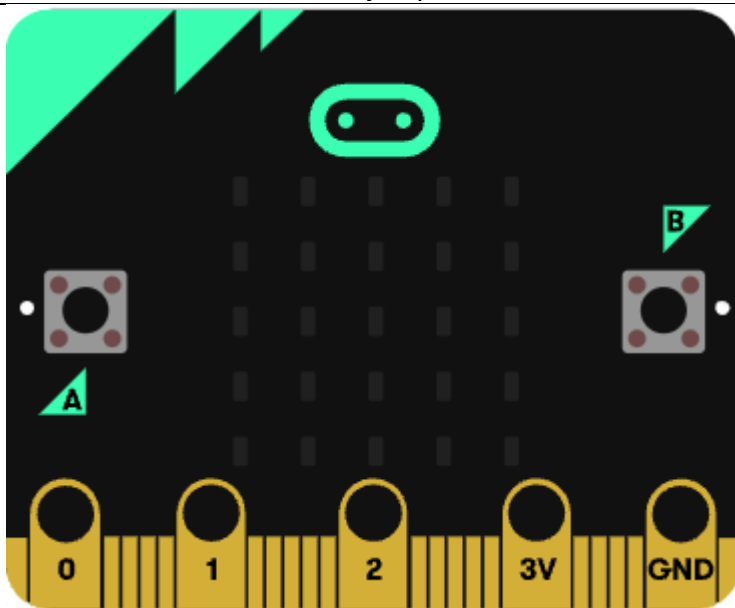
- a) Tento systém rozsvít s nižším jasnem než různé body Z a D , které libovolně umístí na přímku p .
- b) Přímka p je dána parametrickým vyjádřením:

$$x = 2 - t$$

$$y = 1 + 2t, t \in \mathbb{R}$$

Přímku p po druhém stisknutí tlačítka **B** rozsvít.

Geometrické objekty v micro:bitu



Jak vypadá zápis kódu pro tlačítko **B** v JavaScript?



Dotazník

Bc. Kristýna Vacková

Tento dotazník slouží jako zpětná vazba k pracovním listům. Na základně odpovědí budou pracovní listy upraveny, aby plně vyhovovaly potřebám žáka.

Dotazník je zcela anonymní.

I. Část

Škola	ZŠ	SŠ
Třída		
Pohlaví	Muž	Žena
Název pracovního listu		

II. Část

1) Jaká úloha se ti líbila nejvíce? (stačí číslo úlohy)

2) Rozuměl jsi každému zadání?

☐ Ano

☐ Ne

3) Pokud jsi nerozuměl nějakému zadání, které to bylo?

A proč?



4) Přišla ti nějaká z úloh těžká?

☐ Ano

☐ Ne

Která?

A proč?

5) Co bys změnil na pracovním listu?

Velmi děkuji za vyplnění dotazníku.
