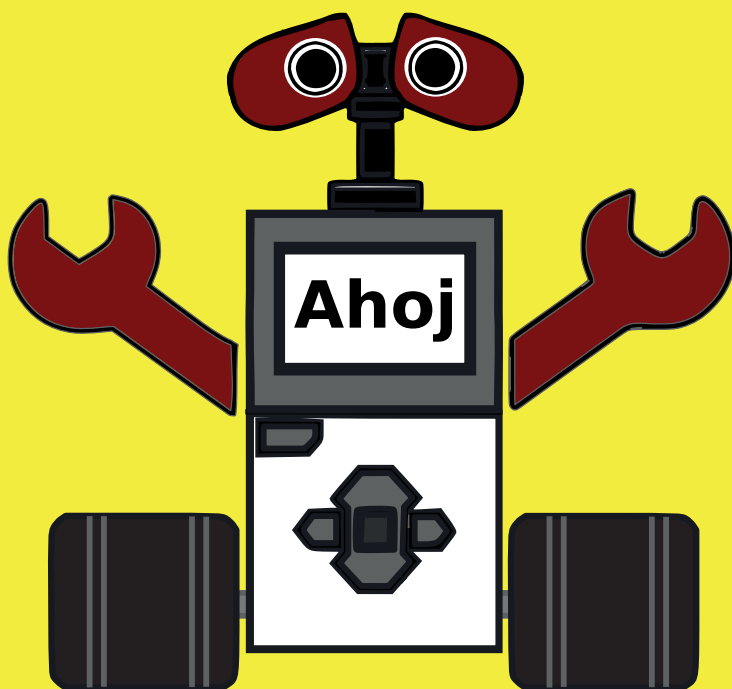


3.DÍL



NEBOJME SE ROBOTŮ

JAROSLAV KOŘÍNEK



2021

Smíchovská průmyslovka třetího věku

3. díl

Nebojme se robotů

Jaroslav Kořínek



Tato brožura vznikla jako inspirační materiál pro volný čas studentů Smíchovské průmyslovky třetího věku. Pěkné čtení i zábavu vám přeje tým SSPŠ a jeho přátelé.

Tato brožura je součástí skupiny následujících brožur:

1. díl: Nanotechnologie pro začátečníky (Věra Krajčová) 
2. díl: Procházka Prahou fyzikální (Jan Mlynář) 
- 3. díl: Nebojme se robotů (Jaroslav Kořínek)** 
4. díl: Magické čtverce (Marie Souchová) 
5. díl: Jedová chýše (Ivan Lukačina) 
6. díl: Fyzikální inspiromat (Věra Krajčová) 
7. díl: Grafika a 3D tisk (Lukáš Vrba) 
8. díl: Procházka Sluneční soustavou (Jan Veselý) 
9. díl: Geometrická zobrazení kolem nás (Tamara Mainzerová) 
10. díl: Kybernetická bezpečnost (Martina Řejhová) 

Grafik a autor loga: Lukáš Chochlovský

Copyright © 2021 – Smíchovská průmyslovka třetího věku

Nebojme se robotů

Co je to robotika a robot?	4
Jak to celé začalo a kam dospělo	5
Zákony robotiky	10
Roboty na doma.....	11
Robot Karel	13
Příkazy.....	13
Pracovní prostředí Karla	16
První složené příkazy pro Karla	18
Závěr	27
Seznam obrázků.....	27
Seznam literatury.....	28

Co je to robotika a robot?

Robotika je vědecká a technická disciplína, která pomáhá člověku řešit jeho problémy s ulehčováním práce a se zvyšováním produktivity práce, a to cestou využití technických prostředků jako výsledků duševní a rozumové činnosti vynálezců, konstruktérů a techniků. Robotika zasahuje do mnoha vědních a technických disciplín a vyžaduje hluboké vzdělání s širokým rozhledem. Úzce souvisí také s elektronikou, mechanikou a výpočetní technikou.

Robotika je předmětem studia na vysokých školách technického zaměření. Jedná se o moderní multidisciplinární obor. Kromě zmíněných souvislostí s elektronikou, mechanikou a softwarem zahrnuje studium i znalosti teorie řízení, měřicí techniky, umělé inteligence a celé řady dalších disciplín. Absolventi tohoto studia nacházejí široké uplatnění prakticky u všech větších firem.

Robot je podle definice stroj pracující s určitou mírou samostatnosti, vykonávající určené úkoly, a to předepsaným způsobem, a při různých mírách potřeby interakce s okolním světem a se zadavatelem. Robot je schopen vnímat své okolí pomocí senzorů, zasahovat do něj, případně si o něm vytvářet vlastní představu.

Ustálená definice pojmu robot doposud neexistuje.

Jak to celé začalo a kam dospělo

Jedním ze základních motivů rozvoje tohoto odvětví jsou bezpochyby přirozené lidské vlastnosti – pohodlnost a lenost.

Jak si ulehčit práci? To je otázka, kterou se zabývá lidstvo od nepaměti. Jedním ze způsobů řešení tohoto problému je přenést svou práci na jiné lidi. Tento princip vévodil celé etapě ve vývoji lidstva – otrokářskému řádu. Uplatňuje se i dnes, na obtížně automatizovatelnou práci najímáme člověka – pracovní sílu – dělníka či řemeslníka. Zaměřme se ale na robotiku, kdy práci člověka zastupuje předem naprogramovaný stroj, což často vede ke zvýšení produktivity práce. Cesta k robotizaci nebyla vždy jednoduchá a přímočará. V podstatě se jednalo o „běh na dlouhou trať“ mnoha vědců a kreativních konstruktérů. Odměnou za jejich pracovitost jim byla prestižní pozice a velmi zajímavá práce jako taková. Tedy docela dobrá příležitost pro chytré, nápadité a tvořivé lidi.

Vývoj mechanizace, automatizace a robotizace byl postupný. Cesta k robotům vedla přes automaty. **Automaty** jsou zařízení, stroje, které automaticky (tedy samočinně) vykonávají určitou operaci, k jejímuž vykonávání byly specificky (jednouúčelově) navrženy a vyrobeny. Za jedno z nejstarších zařízení, které ulehčovalo lidem práci a zvyšovalo produktivitu, můžeme jmenovat tkalcovský stav. Moderními automaty jsou například svářečský automat, výtah, plnicí linka v pivovaru, autopilot, regálový zakladač, automatická pračka či myčka nádobí. Masovým nasazením automatů se podstatně zvýšila produktivita práce a zmenšila se závislost na lidské pracovní síle. A ve výrobě se zvýšila přesnost,

snížila se zmetkovitost. Na nejvyšší příčce v klasifikaci automatů stojí tzv. NC stroje (NC= numeric control – číslicově řízený), což jsou již programovatelná zařízení, která zvládají celou skupinu úloh. Změna sortimentu (výrobku) se řeší přechodem na jiný program.

Roboti-manipulátoři tvoří svébytnou kategorii. Z pohledu vykonávané činnosti by se spíš mohli řadit mezi programovatelné automaty, ale jejich dovybavení senzory pro orientaci v prostředí (např. pro zamezení srážky s neočekávanou překážkou) z nich činí něco, co je opravňuje používat označení robot.

Zásadním mezníkem byl v robotice rok 1920. Poprvé se totiž objevilo slovo robot, a to ve hře Karla Čapka *R.U.R.* Slovo robot je tak dnes nejnámějším českým slovem na světě. Ve 20. století se začaly objevovat první praktické aplikace, které spadají do oblasti robotiky, tzv. **teleoperátory**. Jedná se o manipulátory s ručním řízením bez řídicího systému ovládané člověkem. Jejich úkolem je násobit sílu, eliminovat nedokonalosti člověka a umožnit práci v nepříznivých podmínkách. Použití teleoperátorů je vhodné zejména při manipulaci s polotovary, v medicíně, v rizikových prostředích, stavebnictví (výškové budovy), vojenství, ve vesmíru, v manipulaci s jaderným materiálem, při haváriích. Poté už šel vývoj velmi rychle. V roce 1949 byl zahájen výzkum numericky řízených obráběcích strojů. V roce 1961 byl uveden do provozu první průmyslový robot UNIMATE u firmy General Motors. V roce 1964 byly otevřeny laboratoře umělé inteligence na Massachusett Institute of Technology (M.I.T.), Stanford Research Institute (S.R.I.) a dalších institucích

v USA. Měly se zabývat mj. využitím umělé inteligence v robotice. V roce 1968 byl na S.R.I. postaven mobilní robot Shakey vybavený viděním.



Obrázek 1: Mobilní robot Shakey

V roce 1969 využila NASA robotickou technologii při přistání Neila Armstronga na Měsíci. V roce 1977 dala do prodeje své velmi zdařilé roboty evropská firma

Nebojme se robotů

ASEA. V téže roce roboti vstoupili do širokého povědomí lidí v podobě filmových postav C-3PO a R2D2 z filmu *Star Wars*. Po roce 1980 začínaly být první průmyslové roboty vybavovány počítačovým viděním, čidly hmatu a dalšími prvky, které zatím spadaly do oblasti výzkumu umělé inteligence. V roce 1995 se objevil první chirurgický robotický systém pro tzv. minimálně invazivní chirurgii. V roce 1997 byl na Marsu vysazen robot Sojourner. V roce 2008 byl v Linatexu (výrobce plastových produktů v Herlevu v Dánsku) instalován první spolupracující robot, tzv. kolaborativní robot, zkráceně **cobot**. Do té doby byla robotická pracoviště během činnosti chráněna proti přístupu člověka do jejich pracovního prostoru. Kolaborativní roboty, jak název napovídá, umožňují spolupráci člověka s robotem.



Obrázek 2: Kolaborativní robot

K robotickému výzkumu patří také výzkum dálkového řízení strojů. Při takovém řízení získává operátor všechny informace o prostředí, ve kterém se robot

pohybuje ve formě vhodné pro smysly člověka (zrak, hmat, sluch, čich, chuť), takže má dojem, že je skutečně v prostředí, které obklopuje robota. Ovládání robota má být dokonalé. Operátor prostě provádí totéž, co by dělal, kdyby opravdu v prostředí byl. Takto řízený robot by mohl významně pomáhat hasičům a záchranářům.



Obrázek 3: Výrobní linka v závodu Škoda Auto

Zákony robotiky

V roce 1942 definoval spisovatel Issac Asimov v povídce *Hra na honěnou* tzv.

Základní zákony robotiky. V roce 1950 vyšla povídka ve známější knize povídek *Já, robot*. Tyto zákony, i když jsou definovány spisovatelem sci-fi literatury, by měl ctít každý výzkumník a konstruktér v oboru robotiky.

1. Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby člověku bylo ublíženo.
2. Robot musí uposlechnout příkazů člověka, kromě případů, kdy tyto příkazy jsou v rozporu s prvním zákonem.
3. Robot musí chránit sám sebe před zničením, kromě případů, kdy tato ochrana je v rozporu s prvním nebo druhým zákonem.

Automatizace, robotizace, plně autonomní výrobní linky, robotická vozítka orientující se v terénu, stále výkonnější výpočetní technika, neustále zdokonalované stroje, řídicí systémy, systémy rozpoznávání, zvyšovaný výkon výpočetních jednotek – stroje stvořené člověkem jsou stále dokonalejší a pronikají snad do všech odvětví lidské činnosti od výroby počínaje, přes medicínu, řízení dopravy až po zábavní průmysl.

Roboty na doma

Postavit si a naprogramovat doma vlastního robota je v dnešní době relativně jednoduchá záležitost. Existuje celá řada elektronických stavebnic, které umožňují roboty sestavit. Jsou to například **Ozoboti**, stavebnice BBC **micro:bit**, **Lego**, **Merkur**, **Arduino**, **Raspberry Pi**, **mBot**, **Bioloid Premium Robot Kit**, **Dobot**, **FoXee** a mnoho dalších.

Protože však koupě těchto stavebnic předpokládá jisté investice, pojďme si s roboty pohrát ve virtuálním prostředí. Můžeme využít počítačových her nebo všemocného internetu. Pro začátečníky je naprosto ideální hra **Kodu** společnosti Microsoft. Naleznete ji v obchodě Microsoftu (Microsoft Store) a je zdarma. Pro mírně pokročilé je skvělá hra **Minecraft**, která má mnoho podob. Firma Microsoft se podílela na vývoji její verze Education. Do této verze je zábavnou a nenásilnou formou zapracována výuka mnoha školních témat.

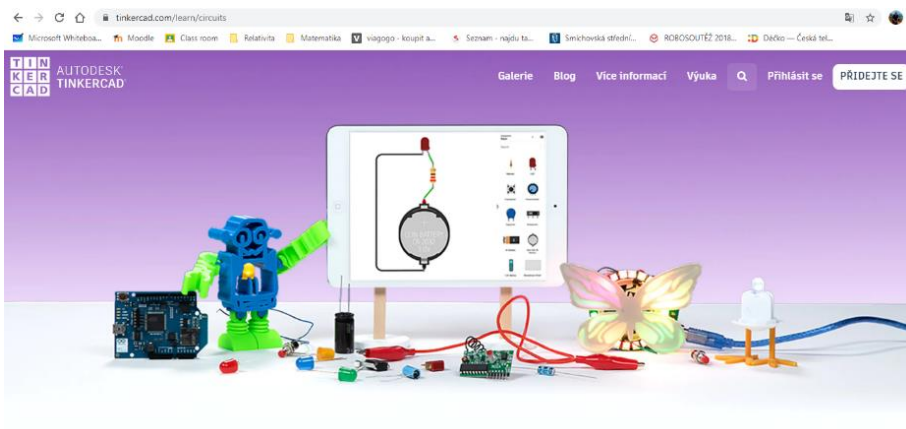


Obrázek 4: Počítačové hry

Další povedenou hrou je hra **LogicBots**, ve které si postavíte robota a toho pak musíte naprogramovat pro zvládnutí zadaných úkolů. Její plnou verzi i demoverzi naleznete na službě Steam. Tato hra již vyžaduje trošičku znalostí z logiky. Na stránkách <http://cyberbotics.com> naleznete vývojové prostředí **Webots**, ve kterém si stavíte roboty a krajinu, ve které se mají pohybovat.

Nebojme se robotů

Tento software však již vyžaduje znalosti programování. Přímo na internetu pak naleznete další vývojová prostředí, ve kterých si můžete roboty vyzkoušet. Například na stránkách tinkercad.com si můžete vyzkoušet 3D modelování, zapojení elektrických obvodů a práci s Arduinem.



Obrázek 5: Webové stránky tinkercad.com

My se blíže podíváme na stránky <http://karel.oldium.net>. Jedná se o prostředí, ve kterém se programuje robůtek **Karel**. Programovací jazyk Karel vytvořil učitel programování, autor prvního programu a knížky *Karel The Robot*, profesor Richard E. Pettis ze Stanfordské univerzity ve Spojených státech. Své jméno získal na počest spisovatele Karla Čapka, autora divadelní hry *R.U.R. – Rossum's Universal Robots*. V bývalém Československu se poprvé objevil v 80. letech v Bratislavě zásluhou doc. RNDr. Jozefa Hvoreckého, CSc. Pro československé mikropočítače (IQ151, PMD85) realizovali své představy o robotu Karlovi Ing. Rudolf Pecinovský, CSc., a Ing. Tomáš Bartovský, CSc.

Robot Karel

Karel je malý, ale šikovný, robůtek, který žije ve svém městě, čtvercovém dvorečku, po kterém se může pohybovat. Může být natočen do jedné ze čtyř světových stran (SEVER, JIH, VÝCHOD, ZÁPAD). Na zádech nosí kouzelný bezedný batůžek, ze kterého může vytahovat značky a klást je na místo, na kterém právě stojí, a opět je sbírat.

Příkazy

Karel umí plnit příkazy. Na začátku zná jen pár základních výkonných a rozhodovacích příkazů. Jeho nejdůležitější vlastností je schopnost učit se nové příkazy. Tyto složené příkazy se Karel učí jako sled již známých příkazů.

Základní výkonné příkazy jsou:

KROK	Udělá jeden krok vpřed. Jít mimo město, stejně jako lézt na zdi, se nedá. Karel sám upozorní, že nemůže příkaz provést.
VLEVO-VBOK	Otočí se o 90° proti směru hodinových ručiček.
POLOŽ	Karel vezme ze svého kouzelného batohu, který má na zádech, značku, a položí ji. Do jednoho políčka se vejde osm značek.
ZVEDNI	Zvedne značku a dá si ji zpátky do batohu.

Rozhodovací a opakovací příkazy:

KDYŽ <i>podmínka</i> <i>příkazy 1</i> KONEC, JINAK <i>příkazy 2</i> KONEC	Karel prozkoumá, jestli je <i>podmínka</i> splněna. Pokud zjistí, že ano, provede <i>příkazy 1</i>, a pokud ne, provede <i>příkazy 2</i>. Uvedení slov KONEC, JINAK následovaných <i>příkazy 2</i> není povinné a může být vynecháno. Ale pozor! Mezera mezi KONEC, JINAK musí být uvedena.
DOKUD <i>podmínka</i> <i>příkazy</i> KONEC	Karel zjistí, jestli je <i>podmínka</i> splněna, a dokud bude platit, bude provádět <i>příkazy</i>.
OPAKUJ <i>počet</i> –KRÁT <i>příkazy</i> KONEC	Karel bude opakovat <i>příkazy</i> tolikrát, kolikrát má uvedeno v parametru <i>počet</i>. Uvedení slova –KRÁT není povinné.

Podmínka je jedno slovo, případně spojení dvou slov. Může to být spojení slova **JE**, které je nepovinné, nebo **NENÍ** a jednoho dalšího slova:

ZEĎ	Jestliže JE/NENÍ před Karlem postavena zeď, pak je podmínka splněna.
ZNAČKA	Jestliže JE/NENÍ na políčku ve městě, kde Karel právě stojí, umístěna značka, pak je podmínka splněna.
DOMOV	Jestliže JE/NENÍ políčko, na kterém Karel stojí, označeno symbolem domečku, Karel se nachází či nenachází doma a podmínka je splněna.
VÝCHOD	Jestliže JE/NENÍ Karel natočen na východ, je podmínka splněna.
SEVER	Jestliže JE/NENÍ Karel natočen na sever, je podmínka splněna.
ZÁPAD	Jestliže JE/NENÍ Karel natočen na západ, je podmínka splněna.
JIH	Jestliže JE/NENÍ Karel natočen na jih, je podmínka splněna.

Kombinací základních, podmíněných a cyklických příkazů můžeme vytvářet příkazy složitější. Jakmile je již jednou nový příkaz vytvořen, můžeme ho použít k tvorbě dalších příkazů.

Pracovní prostředí Karla

Věnujme se nyní popisu pracovního prostředí. Stránka je rozdělena na 4 sekce. První tvoří horní řádek se jménem autora projektu, ale především s odkazy na nápovědu a příklady. Tyto dva odkazy v levém horním rohu jsou naprosto zásadní. Pomocí nich se dá ovládání celého prostředí jednoduše zvládnout.

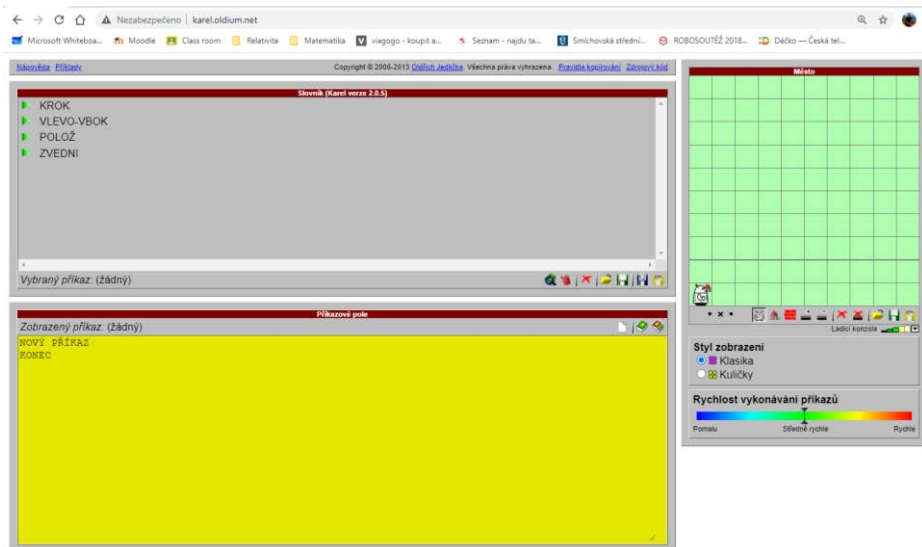
V pravé části je tzv. město. Jedná se o čtvercové pole 10×10 políček, kde se může Karel pohybovat. Do políček lze ukládat značky nebo stavět zdi. Zdí se také rozumí ohraničení města. Dále je možné v tomto okně nastavit, zda se budou značky zobrazovat jako kuličky nebo jako čárky. V panelu „rychlost vykonávání příkazů“ lze nastavit rychlost pracovní činnosti.

Horní velké okno vlevo, které se nazývá slovník, zobrazuje všechny výkonné příkazy, které Karel zná. Jejich seznam se bude rozšiřovat tak, jak budete vytvářet nové příkazy. Zelená šipka vlevo od příkazu tento příkaz aktivuje a Karel jej provede.

Žluté okno s názvem příkazové pole umožňuje psát nové příkazy nebo je editovat.

Je důležité si uvědomit, že po zavření těchto webových stránek o veškerou práci přijdete. Jejich opětovným otevřením se načte nový čistý projekt. Pokud si chcete práci uložit, je potřeba tak učinit pomocí ikony zelené diskety v okně

Slovník. Tím se otevře okno, v jehož zelené části se vygeneruje posloupnost znaků. Tuto celou větu je potřeba označit (např. kombinací kláves CTRL+A) a zkopírovat do schránky (CTRL+C). Následně je potřeba ji vložit do libovolného textového editoru (Word, Notepad...) a tento soubor si pak uložit. Asi nejjednodušší metoda je přímo na ploše kliknout pravým tlačítkem a z menu si vybrat volbu Nový – Textový dokument. Ten si pojmenovat, poklepáním otevřít a vložit do něj pomocí CTRL+V zkopírovaný text. Naopak, chcete-li si nahrát již uloženou práci, zkopírujete kompletní text v uloženém souboru, stisknete v okně slovník ikonu žlutého šanonu a do otevřeného okénka zkopírovaný text vložíte. Stejným způsobem se ukládá nebo načítá město, které si vytvoříte. V okně město jsou ikony stejného významu. Pro uložení kompletní práce musíte uložit jak slovník, tak město.



Obrázek 6: Prostředí robota Karla

První složené příkazy pro Karla

Vytváření nových příkazů není nic složitého. Ukažme si například příkaz pro vpravo v bok. Karel na začátku umí jen vlevo v bok. Otočit se na druhou stranu pro něj tedy znamená provést třikrát příkaz VLEVO-VBOK. Cítíte sami, že psát vždy třikrát ten samý příkaz, je nepraktické. Vyplatí se tedy vytvořit příkaz nový.

Ve žlutém okně klepneme vpravo na ikonu bílého listu nový příkaz. Na prvním řádku přepíšeme text NOVÝ PŘÍKAZ na požadovaný název, např. VPRAVO-VBOK, a stiskneme enter. Na další řádek napíšeme požadovaný příkaz (Karel ho již musí znát!). Doporučuji pro lepší přehlednost pomocí mezerníků příkazy odrážet na stejnou úroveň. Na každém řádku je vždy jeden příkaz. Náš příkaz pro otočení vpravo by tedy mohl vypadat takto:

```
VPRAVO-VBOK
  VLEVO-VBOK
  VLEVO-VBOK
  VLEVO-VBOK
KONEC
```

Po dopsání je potřeba příkaz uložit stisknutím ikony zelené knížečky. V tu chvíli se příkaz objeví v horním okně slovník a můžeme ho používat. Pro zajímavost ukážeme ještě další možnost, jak by mohl příkaz vypadat s využitím cyklu:

```

VPRAVO-VBOK-JINAK
    OPAKUJ 3-KRÁT
        VLEVO-VBOK
    KONEC
KONEC

```

Podobným způsobem můžeme například vytvořit příkaz pro čelem vzad:

```

ČELEM-VZAD
    OPAKUJ 2-KRÁT
        VLEVO-VBOK
    KONEC
KONEC

```

Máme-li vytvořen příkaz ČELEM-VZAD, můžeme vytvořit příkaz pro vpravo v bok pomocí tohoto příkazu:

```

VPRAVO-VBOK-JEŠTĚ-JINAK
    ČELEM-VZAD
        VLEVO-VBOK
    KONEC

```

Vyzkoušejte si funkčnost příkazů jejich spuštěním kliknutím na zelenou šipku vlevo od nich. Abyste lépe viděli chování Karla, můžete si ho postavit kamkoliv na šachovnici. Klepněte na ikonu Karla pod zelenou šachovnicí, pak klepněte levým tlačítkem myši na nové políčko, kam ho chcete přemístit. Provedete-li přímo na Karlovi rychlý dvojklik, provede otočení vlevo. Spuštěním příkazu VPRAVO-VBOK je vidět, jak Karel vykoná postupně 3 pootočení vlevo. Pokud vám to přijde pomalé, zkuste v posuvném panelu pod městem nastavit vyšší rychlost vykonávání příkazů. Nastavení rychlosti má význam při ladění

Nebojme se robotů

složitějších příkazů, aby bylo zřetelně vidět, co Karel skutečně dělá. Při velké rychlosti nemusí být jeho činnost úplně zřejmá.

Ukažme si další jednoduchý příklad. Nechme Karla jít tak dlouho, dokud nenalezne zeď. K tomu budeme potřebovat podmíněný cyklus. Nazvěme příkaz např. KROKY:

```
KROKY
DOKUD NENÍ ZEDĚ
KROK
KONEC
KONEC
```

Karel půjde tak dlouho, dokud nenarazí na zeď. Připomínám, že zdí se rozumí ohrazení šachovnice nebo symbol zdi. Zeď na šachovnici umístíte tak, že klepnete na ikonu cihlové zdi pod městem a pak klepnete na políčko, kam chcete zeď postavit. Opětovným klepnutím na zeď na šachovnici ji odeberete. Zdi můžete přidávat a odebírat, i když Karel vykonává příkaz.

Zkusme dát Karlovi úkol, ať se otočí vlevo, když narazí na zeď, a pokračuje dál v chůzi k další zdi, u které se opět otočí vlevo a tak dále. Tentokrát si ukážeme použití podmínky. Nazvěme ho CHŮZE:

```
CHŮZE  
  KDYŽ NENÍ ZEĎ  
    KROK  
  KONEC, JINAK  
    VLEVO-VBOK  
  KONEC  
  CHŮZE  
  KONEC
```

Tento příkaz využívá navíc tzv. rekurze neboli volá sám sebe. Jak to vlastně funguje? Vytvořili jsme nekonečný příkaz. Poté, co Karel buď udělá krok, nebo se otočí vlevo, je opět zavolán program CHŮZE. Takto se vlastně ocitá v nekonečné smyčce. Program ukončíme stisknutím červeného křížku vlevo vedle příkazu. Zelená šipka se změnila na křížek, což znamená, že Karel vykonává nějaký příkaz. Během jeho putování mu zkoušejte dávat do cesty zdi a sledujte jeho chování.

K ukončení jeho činnosti můžeme využít například domečku, který je součástí města. Využijeme k tomu opět podmínky. Příkaz nazveme CHŮZE-DOMŮ. Ukažme si, jak využít již hotový příkaz k vytvoření příkazu nového, ve kterém je jen pár drobných změn. Klepneme v okně slovník na příkaz CHŮZE (podbarví se zeleně). V pravé dolní části okna klepneme na první ikonu – lupa . Tím se nám příkaz zobrazí dole ve žlutém okně příkazové pole. Opravíme první řádek na název CHŮZE-DOMŮ a dolů přepíšeme další podmínku, viz níže:

```
CHŮZE-DOMŮ  
KDYŽ NENÍ ZEŘ  
KROK  
KONEC, JINAK  
VLEVO-VBOK  
KONEC  
KDYŽ JE DOMOV  
STOP  
KONEC, JINAK  
CHŮZE-DOMŮ  
KONEC  
KONEC
```

Ovšem nyní pozor! Vytváříme nový příkaz, takže ho také musíme jako nový uložit . Kdybychom použili prostřední ikonu zelené knížky , přepsaly bychom původní příkaz CHŮZE!

Vyzkoušejte si během Karlova putování vkládání zdí do jeho cesty a změnu umístění domečku. Vhodným vkládáním zdí nasměrujte Karla domů. Domeček umístíte jinam ve městě tak, že klepnete na ikonu domečku pod městem  a pak klepnete na místo ve městě, kam chcete domeček umístit.

Jako další příklad si ukážeme umísťování značek na políčko ve městě. Nechme například Karla při procházce městem položit na políčko, které navštívil, značku. Upravíme si příkaz CHŮZE na nový příkaz CHŮZE-S-KLADENÍM:

```

CHŮZE-S-KLADENÍM
  KDYŽ NENÍ ZEĎ
    KROK
    POLOŽ
  KONEC, JINAK
    VLEVO-VBOK
  KONEC
  CHŮZE-S-KLADENÍM
  KONEC

```

Nechám-li Karla chodit dostatečně dlouhou dobu, po čase program havaruje, neboť Karel nemůže položit na jedno políčko více jak 8 značek. Příkaz můžeme upravit například na tvar, kdy značku položí pouze tehdy, je-li políčko prázdné:

```

CHŮZE-S-KLADENÍM
  KDYŽ NENÍ ZEĎ
    KROK
    KDYŽ NENÍ ZNAČKA
  POLOŽ
  KONEC
    KONEC, JINAK
      VLEVO-VBOK
    KONEC
  CHŮZE-S-KLADENÍM
  KONEC

```

Anebo ho můžeme upravit tak, aby pro změnu značku sebral, když na políčku bude:

CHŮZE-S-KLADENÍM

KDYŽ NENÍ ZEĎ

KROK

KDYŽ NENÍ ZNAČKA

POLOŽ

KONEC, JINAK

ZVEDNI

KONEC

KONEC, JINAK

VLEVO-VBOK

KONEC

CHŮZE-S-KLADENÍM

KONEC

Dejme nyní Karlovi úkol, ať přemístí všechny značky z konkrétního políčka o jedno políčko dál. Sled příkazů bude jednoduchý. Dokud bude na konkrétním políčku nějaká značka, sebere ji, udělá krok vpřed, položí značku, udělá čelem vzad, krok zpět na původní políčko, čelem vzad a opět, pokud, tam je značka, ji sebere a pokračuje tak stále dokola, než dojdou značky. Příkaz nazvěme PŘESUN:

```

PŘESUN
DOKUD JE ZNAČKA
  ZVEDNI
  KROK
  POLOŽ
    ČELEM-VZAD
  KROK
  ČELEM-VZAD
KONEC
KROK
KONEC

```

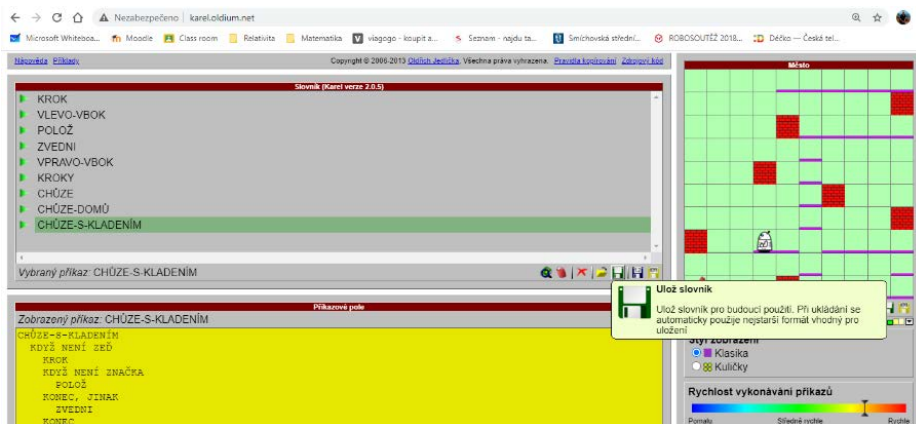
Než příkaz spustíte, připravte si ve městě políčko s několika značkami a na něj umístěte Karla. Před Karlem musí být pochopitelně volné políčko. Chcete-li vymazat již město zaplněné značkami a cihlami, stačí stisknout tlačítko s ikonou červeného kříže , které je v řadě tlačítek pod městem. Podobně lze dalším tlačítkem  vymazat pouze všechny značky.

Při této činnosti se Karel dost naběhá. Pro každou značku se musí vrátit a přemístit ji samostatně, a přitom má bezedný batoh. Co kdybychom to zařídili tak, že posbírá všechny značky a pak je zase najednou všechny vyloží na nové políčko. To opravdu jde. Využijeme k tomu opět rekurze. Toto řešení uvádíme již jen jako třešničku na dortu. Pochopení asi není úplně triviální. Důležité pro nás je, že část za rekurzivním voláním se vykoná stejně krát jako část před rekurzí. Vyzkoušejte a uvidíte. Příkaz nazveme PŘESUN-NAJEDNOU:

Nebojme se robotů

```
PŘESUN-NAJEDNOU
      KDYŽ JE ZNAČKA
ZVEDNI
PŘESUN-NAJEDNOU
      POLOŽ
KONEC, JINAK
      KROK
      KONEC
KONEC
```

S robůtkem Karlem se dá zažít opravdu hodně zábavy. Je jen na vaší fantazii, jaké úlohy Karlovi vymyslíte. Zkuste například položit značku na každé pole ve městě nebo je položte pouze na úhlopříčky. Zkuste Karlovi postavit bludiště ze zdí a nechat ho jím procházet. Nechte se inspirovat ze záložky příklady. Zobrazené příkazy stačí zkopírovat do příkazového pole a rovnou je používat. Na závěr připomínáme, že práci je potřeba si uložit!!! Zavřením webového prohlížeče o ni přijdete!



Obrázek 7: Prostředí robota Karla – uložení práce

Závěr

A to je pro začátek vše. V případě zájmu můžeme vytvořit další zajímavé modely a programy.

Seznam obrázků

Obrázek 1: Mobilní robot Shakey	7
Obrázek 2: Kolaborativní robot.....	8
Obrázek 3: Výrobní linka v závodu Škoda Auto	9
Obrázek 4: Počítačové hry	11
Obrázek 5: Webové stránky tinkercad.com.....	12
Obrázek 6: Prostředí robota Karel	17
Obrázek 7: Prostředí robota Karel – uložení práce.....	26

Seznam literatury

Novák, P.: Mobilní roboty – pohony, senzory, řízení. Praha, BEN, 2005, ISBN 80-7300-141-1.

Maršík, A., Kubičík, M.: AUTOMATIZACE – automatické řízení ve strojírenství. Praha, SNTL, 1980, typové číslo L26-C2-I-01/55536.

Roboty a manipulátory. Učební text VOŠ a SPŠ Kutná Hora

Šolc, F., Žalud, L.: Robotika. Skripta VUT Brno

Hlaváč, V.: Úvod do robotiky. Prezentace. FEL ČVUT, katedra kybernetiky, Centrum strojového vnímání

Osoba, J.: PC-Karel 2001. www.pckarel.sweb.cz, Copyright: © 1996-2004, Jiří Osoba. All Rights Reserved.

www.karel.oldium.net. Copyright: © 2006-2013, Oldřich Jedlička. Všechna práva vyhrazena

