evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Programovací jazyk Karel – úvod

Číslo projektu	CZ.1.07/1.5.00/34.0950
Kódování materiálu	vy_32_INOVACE_inf3_prg02
Označení materiálu	prg02_karel_uvod
Název školy	Gymnázium Kladno
Autor	Mgr. Pelikánová Lucie
Anotace	Výukový materiál Algoritmizace a programování, materiál je určen pro LMS systém, lze jej využít také pro samostatnou práci žáků. Tato část kurzu seznamuje s vývojovým prostředím programovacího jazyka Karel (speciální úprava).
Předmět	Informatika a výpočetní technika
Tematická oblast	Algoritmizace a programování
Téma	Programovací jazyk Karel – úvod
Očekávané výstupy	Žák se seznámí s vývojovým prostředím, základními příkazy jazyka, s ovládáním robota Karla. Ovládne tvorbu vlastních příkazů, vytvoří několik vlastních nových příkazů. Uvědomí si základní principy strukturovaného programování.
Klíčová slova	Programovací prostředí Karel, základní příkazy, cyklus, strukturované programování, repeat.
Druh učebního materiálu	výukový kurz (lekce)
Ročník	3
Cílová skupina	vyšší stupeň osmiletého gymnázia, čtyřleté gymnázium
Ověřeno	7. 5. 2013, O7
Zdroje uvedeny na konci dokumentu.	

Metodický pokyn

Vstupní soubory a složky:

Karel	programovací jazyk Karel, včetně manuálu
prg02_karel_uvod.pdf	výukový text
prg02_uvod.krl	praktické ukázky

Výstupy (řešení):

Tato ukázková řešení nejsou určena pro žáky, ale pro učitele.

reseni/prg02_uvod_reseni.krl	řešení zadaných úloh

Téma: Programovací jazyk Karel – úvod

Manuál k programu

Kompletní originální manuál je součástí jazyka a je uložen ve složce Karel spolu s instalací programu (soubor karel_manual.doc).

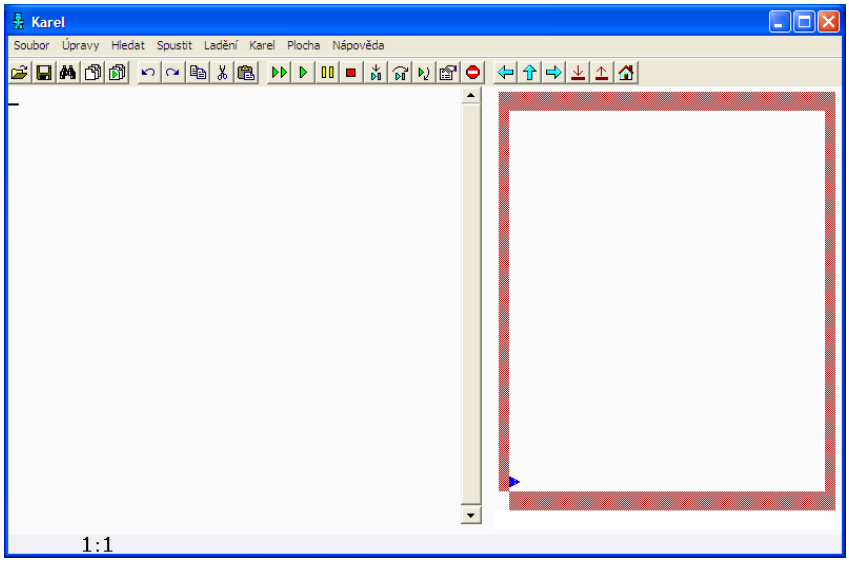
Program Karel je určen na základní výuku programování a algoritmizace. Pomocí něj se hravě seznámíte se základními druhy příkazů, které se používají v programovacích jazycích. Jednoduché prostředí Vám umožní soustředit se na problém.

Karel je jméno robota, který umí vykonávat příkazy podle programů, které mu napíšete. Graficky je znázorněn do podoby šipky, ukazující směr, kterým je natočen. Jeho základní příkazy, které zná, jsou velmi omezené:

`step (krok)`	vykoná krok ve směru šipky
`left (doleva)`	otočí se o 90° vlevo
`right (doprava)`	otočí se o 90° vpravo
`put (polož)`	položí cihličku (zelená čárka)
`pick (zvedni)`	zvedne cihličku
`home (domů)`	Karel se přemístí do levého rohu plochy, otočený k východu

Příkazy si můžete vyzkoušet pomocí tlačítek, které jsou umístěny nad plochou (místností) ve které se Karel pohybuje.

Vývojové prostředí Karla



Do levé části okna píšete a upravujete programy (procedury), kterými učíte Karla nové příkazy. Vpravo se nachází místnost, po které se Karel pohybuje (její velikost je možné změnit). Nad místností jsou ikony pro ovládání robota.

Praktické cvičení č. 1

1. Vyzkoušejte, jak lze zadávat Karlovi základní příkazy
2. Kolik cihliček může položit?
3. Co se stane, když pokládáte cihličku, která se již nevejde?
4. Můžete použít příkaz pick, pokud již není pod Karlem žádná cihlička?

Chybové hlášení

Pokud nutíte dělat Karla něco, co již není možné (udělat krok do zdi, položit cihlu, která se již nevejde apod.), objeví se **chybové hlášení**. Pokud se toto hlášení objeví při realizaci programu, znamená to, že program skončil s chybou a nemůže pracovat dál. Tato situace by nastat neměla.

První programy

Z čeho se skládá program. Karel je jazyk strukturovaný. Program se skládá z menších celků – procedur. Pomocí nich učíte Karla nové příkazy.

Dvojkrok

Naučte Karla udělat 2 kroky najednou. Pojmenujte tuto novou proceduru dvojkrok. Pravidla pro psaní názvů procedur: velká a malá písmena (včetně písmen s háčky nebo s čárkami), číslice a také znaky ? ! _ : ` @ # ... První znak názvu nesmí být číslice.

Prvním řádkem procedury je slovo **procedure**, za kterým následuje název procedury. Procedura je ukončena příkazem **end**.

```
procedure dvojkrok
  step
  step
end
```

Program se spouští magnetofonovým tlačítkem nebo stiskem kláves F9 (rychle) a F6 (pomalu). Při pomalém spuštění vidíte přímo animaci Vašeho programu. Spustí se ta procedura, ve které máte umístěn kurzor (nemusí být na začátku.

Otočka o 180 stupňů

Naučíme Karla příkaz **otoc**, kterým se otočí o 180 stupňů.

```
procedure otoc
  left
  left
end
```

Každý příkaz píšeme raději na jednu řádku. Je možné psát více příkazů na řádku, ale musíme je oddělit středníkem.

Odsazujte příkazy na stejné úrovni od kraje. To není kvůli Karlovi, ale kvůli Vám nebo někomu, kdo po Vás program bude číst. Odsazení přispěje k lepší čitelnosti celého kódu.

Praktické cvičení č. 2

1. Naprogramujte oba nové příkazy, zkuste je spustit.
2. Jakým jiným způsobem by bylo možné Karla ještě otočit?
   Vyzkoušejte, novou proceduru pojmenujte jako otoc2.
3. Vše uložte, soubor s řešením dostane automaticky koncovku **krl**.
4. Naprogramujte příkaz **tam_a_zpet** (pro název s mezerou použijte podtržítka, název musí tvořit celé slovo). Karel udělá dva kroky jedním směrem, otočí se a udělá dva kroky zpět. Využijte k tomu již naprogramované procedury **dvojkrok** a **otoc**.

Na konci dokumentu je **Řešení** úloh, podívejte se, a pokud máte řešení takto, pochopili jste strukturované programování.

Jak neprogramovat:

```
procedure takto_ne
  step
  step
  left
  left
  step
  step
end
```

Řešené programy také naleznete v souboru prg02_uvod.krl.

Cyklus s pevným počtem opakování

Vyplnění pole místnosti cihlami

Do každého pole místnosti se vejde 9 cihel. Pomocí cihel můžete v místnosti vytvářet „stavby“. Nejprve vyzkoušíme vyplnit jedno pole cihlami. Pokud bych ho nechtěla mít plné, mohla bych umístit např. jen 5 cihel. Jak to lze udělat:

Jednoduché, ale nevhodné řešení

```
procedure vypln_napul_primitivni
  put
  put
  put
  put
  put
end
```

Představte si, že až budete chtít vyplnit celé políčko, musíte psát příkazy devětkrát. To byste měli opravdu dlouhé programy. Lepší je použít cyklus. Cyklus má začátek a konec a příkaz, který je uvnitř, se opakuje.
Začátek cyklu: **repeat *kolikrát***
Konec cyklu: **end**

Vypln_napul

```
procedure vypln_napul
//cyklus
  repeat 5
```

```
    put
  end //od repeat
end // end od procedury
```

Text za // (dvojitým lomítkem) je komentář. Je vhodné do programů psát, co se děje.

Praktické cvičení č. 3

Naprogramujte proceduru vypln, která vyplní celé pole cihlami. Naprogramujte samostatně ☺. Řešení naleznete v řešení.

Jak obnovit celou plochu

Příkazem F5 nebo výběrem příkazu Obnovit v nabídce Plocha, vymažete celou plochu a Karla umístíte domů (do levého spodního rohu).

Stavba zdi

Karel postaví řadu cihel (celých vyplněných polí). Nejprve vytvoříme příkaz **kratka_zed**, která bude tvořena čtyřmi vyplněnými poli.

Krátká zeď

```
procedure kratka_zed
  repeat 4
    vypln
    step
  end
end
```

Praktické cvičení č. 4

Dlouhá zeď má délku 8 polí. Můžete soutěžit o nejkratší kód pro realizaci dlouhé zdi. Naprogramujte. Řešení není ani v řešení ☺

Řešení

(2) Úloha tam_a_zpet

```
procedure tam_a_zpet
  dvojkrok
  otoc
  dvojkrok
end
```

(3) Vypln

```
procedure vypln
  repeat 9
    put
  end
end
```

Zdroj

[1] Laštovička Petr, http://petr.lastovicka.sweb.cz autor programu a základního manuálu. (program má free licenci k užívání a současně mám souhlas autora s rozšiřováním a používáním).