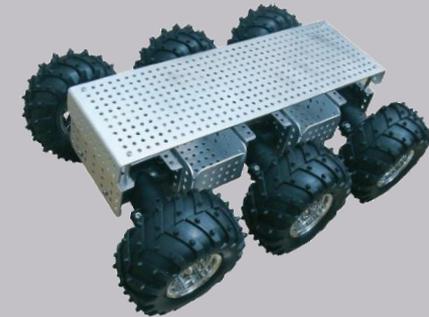


ZÁKLADY PROGRAMOVÁNÍ

Mgr. Vladislav BEDNÁŘ

- 2014
- 5.9
- 10/14



Co je vhodné vědět, než si vybereme programovací jazyk a začneme programovat roboty.

ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Programovací jazyky

- Programovací jazyky jsou jazyky sloužící k tvorbě počítačových programů (programování). Programování je proces algoritmizace dané úlohy, tj. vytváření postupu, jenž vede k řešení dané úlohy.
- Programovací jazyky se dají rozdělit podle mnoha kritérií. Nejčastější způsoby jsou na vyšší a nižší.
- Nižší programovací jazyky jsou jazyky primitivní, jejichž instrukce (více méně přesně) odpovídají příkazům procesoru



0:35

2 / 31

0:40



Nižší programovací jazyky

- Patří sem jazyk symbolických adres (Assembler) a strojový kód.
- V těchto jazycích musí programátor vypisovat každou maličkost, i jednoduchý program má neúměrně složitý zdrojový kód. Výhodou je, že programátor má takto přístup i k funkcím počítače, které by měl ve vyšším programovacím jazyce nedosažitelné. Zvláštním typem nižšího jazyka je tzv. autokód, který spojuje prvky nižších a vyšších jazyků. Vznikl rozšířením Assembleru o příkazy pro často používané skupiny instrukcí.

Vyšší (problémově orientované) jazyky

- Struktura zdrojových kódů těchto jazyků je logická, nejsou závislé na strojových principech počítače. Do strojového kódu se převádějí kompilátorem (případně se rovnou spouštějí interpretrem). V praxi je vyšší programovací jazyk vše, co není Assembler, to znamená: Pascal, Basic, Prolog, Lisp, Algol, Fortran atd.
- Často se uvádí, že jazyk C je jakýmsi přechodem mezi vyššími a nižšími jazyky, má však blíže k vyšším.

ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Imperativní, logické a funkcionální jazyky

- Imperativní (též procedurální) jazyky jsou téměř všechny jazyky, které se běžně používají. K řešení úlohy se používá algoritmu (postupu, jak se má daná úloha vyřešit). Např. Pascal, C, Basic, PHP, Perl, Java.
- U logických jazyků programátor pouze popíše daný problém pomocí logických výroků. Program z nich potom vyvozuje požadované informace.



0:35

5 / 31

0:40



Interpretované a kompilované jazyky

- Interpretované jazyky jsou překládány až za běhu programu. Hlavní nevýhodou těchto jazyků je, že se musejí vždy spouštět v interpretru. Do této skupiny patří většina verzí Basicu, všechny skriptovací jazyky (PHP, Python).
- Kompilované jazyky jsou celé přeloženy a až potom mohou být spuštěny. Jsou rychlejší, mají vyšší nároky na formální správnost kódu. Překládají se kompilátorem, výsledkem překladu je (většinou) exe soubor.

Jazyk Pascal

- V dnešní době začíná Pascal (a celkově strukturované programování) poněkud zastarávat a objevují se snahy nahradit jej na pozici učebního jazyka moderním objektově orientovaným Pythonem.

```
Program Program1;  
Var I : Integer;  
Begin  
  writeln('Vitam te u sebe.');
```

For I := 1 to 7 do

```
  Begin  
    write(I);  
    writeln(' Bedy te vita.');
```

End; end.

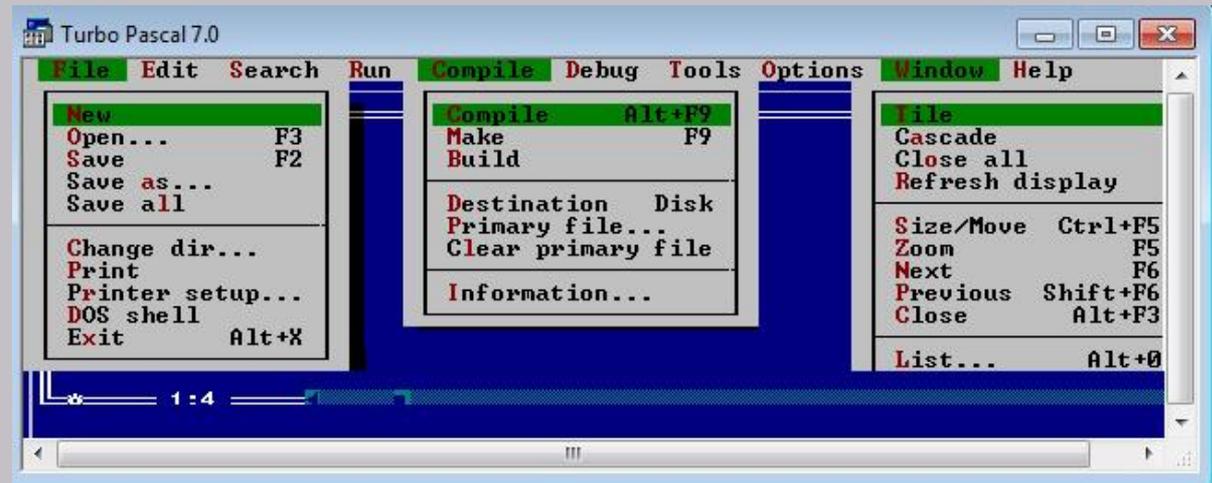
ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU

Turbo Pascal

- Turbo Pascal (Borland Pascal) je nejrozšířenější implementací programovacího jazyka Pascal.

- Vývojové prostředí Turbo Pascalu integruje textový



editor (pro pořizování a úpravy zdrojových textů), kompilační překladač, linker a debugger (ladicí prostředek)

Jazyk C

- Programovací jazyk C je známý svou přenositelností a rychlostí. Byl navržen jako poměrně malý jazyk, kombinující efektivitu a výkonnost. Byl napsán pro operační systém UNIX. C je ryze procedurální jazyk, to znamená, že zdůrazňuje algoritmy.

```
#include <stdio.h>
void main()
{
    printf("Vitam te u sebe.\n");
    for(int i=1; i<6; i++)
    {
        printf("%d Bedy te vita.\n", i);
    }
}
```

Jazyk C++

- C++ je objektově orientovaný jazyk, to znamená, že zdůrazňuje data. Myšlenka OOP spočívá v návrhu datových tříd, které popisují určitý souhrn vlastností a jak s těmito vlastnostmi zacházet.

- Má velké množství knihoven.

```
#include <iostream>
using namespace std;
int main ()
{
    cout << "Vitam te u sebe.\n";
    for(int i=1; i<6; i++)
    {
        cout << i << " Bedy te vita.\n";
    }
    return 0;
}
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Jazyk C++ a vývojové prostředí

- Jedním z množství integrovaných vývojových prostředí je Dev-C++ pro programovací jazyk C / C + + .
- Obecně se má, zato, že C/C++ jsou jazyky složité.

```
File Edit Search View Project Execute Debug Tools Window Help
[Icons] Compile Ctrl+F9 Run F9
[Icons] Compile & Run Ctrl+F10 Rebuild All Ctrl+F11 Clean
Main.c
Main.h
Main.c
Main.h
MyWindowClass;
Inst = NULL;

#define IDC_MAIN_TEXT 1001

BOOL LoadFile(HWND hEdit, LPSTR pszFileName)
{
    HANDLE hFile;
    BOOL bSuccess = FALSE;

    hFile = CreateFile(pszFileName, GENERIC_READ, FILE_SHARE_READ, NULL,
        OPEN_EXISTING, 0, 0);
    if(hFile != INVALID_HANDLE_VALUE)
    {
        DWORD dwFileSize;
        dwFileSize = GetFileSize(hFile, NULL);
        if(dwFileSize != 0xFFFFFFFF)
        {
            LPSTR pszFileText;
            pszFileText = (LPSTR)GlobalAlloc(GPTR, dwFileSize + 1);
        }
    }
}
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU

Basic

- Jeho typickou vlastností bylo číslování řádků. Instrukce se prováděly od řádku s nejmenším číslem až po řádek s největším číslem. Byla možná takzvaná adresace řádku. Basic byl zaveden jako jednoduchý nástroj pro výuku programování.

01 CLS

02 PRINT "Vitam te u sebe"

03 PRINT "Stisknete cokoliv pro

pokracovani"

05 PAUSE

06 CLS

10 FOR a=1 TO i-

15 PRINT "Bedy te vita.",i

20 PLOT 1+(p(2,a)-xmin)*rx,

1+(p(1,a)-ymin)*ry

21 DRAW rx*(p(2,a+1)-p(2,a)),

ry*(p(1,a+1)-p(1,a))

30 NEXT a

Visual Basic - NET Framework

- Vývoj se zastavil ale jeho poslední verze byla koncipována pro tvorbu programů v prostředí NET Framework. NET Framework je prostředí spustitelné v operačním systému Windows nutné pro běh programů napsaných v jakémkoli jazyce pro NET.

```
Module Module1
  Sub Main()
    Console.WriteLine("Vitam te
u sebe.")
    For i As Integer = 1 to 5
      Console.WriteLine("{0}
      Bedy te vita.",i)
    Next
  End Sub
End Module
```

ROBOT

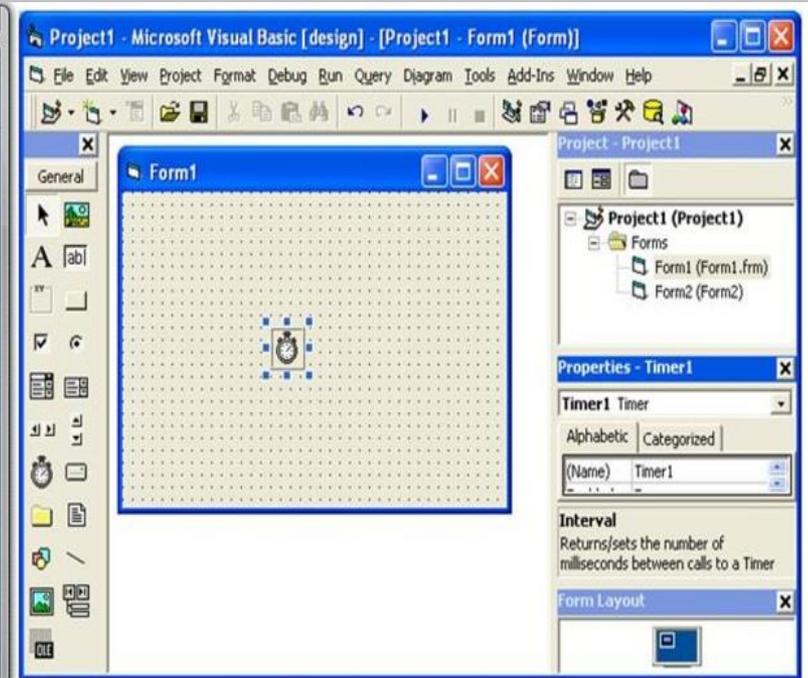
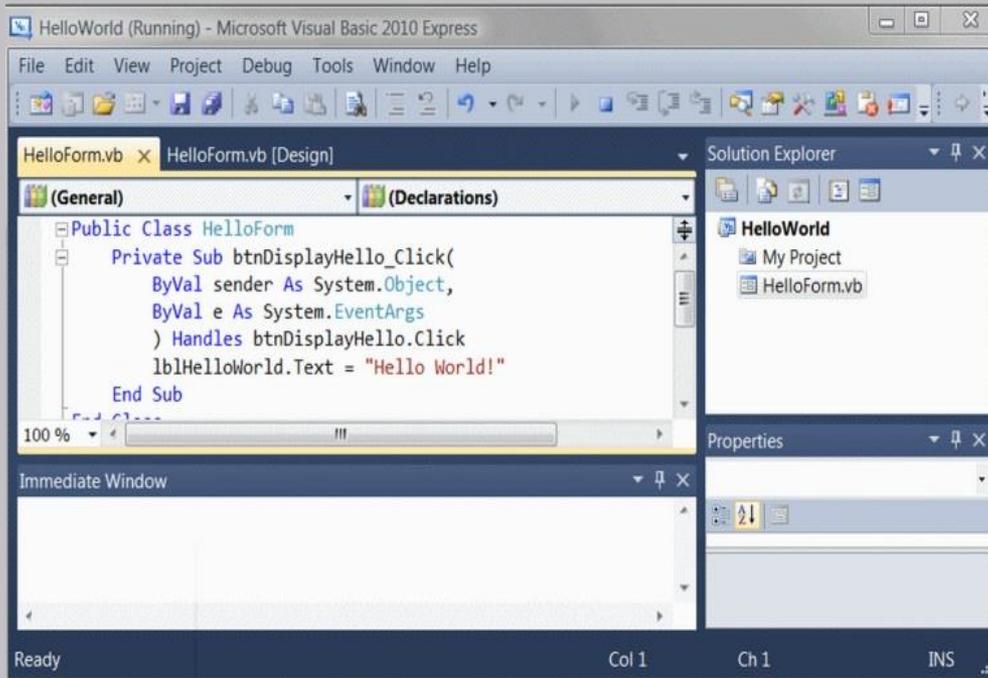
VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Visual Basic

- Od firmy Microsoft pochází Visual Basic 2008 Express Edition.



C # (C SHARP)

- C# je vlastně vylepšená a zjednodušená objektová verze programovacího jazyka C++. Nezbytnou podmínkou pro programování v jazyce C# je prostředí NET Framework.

```
using System;  
class Program  
{  
    static void Main()  
    {  
        Console.WriteLine("Vitam te u sebe.");  
        for(int i=1; i<6; i++)  
        {  
            Console.WriteLine("{0} Bedy te vita.",i);  
        }  
    }  
}
```

ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU

Perl

- Perl je interpretovaný programovací jazyk. Nemusíme zde kompilovat samostatně. Program je zkompilován po každém spuštění automaticky a můžeme kdykoli „přikompilovat“ další kód.

```
#!/usr/bin/perl
```

```
print "Vitam te u sebe.;"
```

```
for ($i = 1; $i < 6; $i++)
```

```
{
```

```
    print "$i Bedy te vita.;"
```

```
}
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

SmallTalk

- Smalltalk je čistý objektově orientovaný jazyk

'Vitam te u sebe.' displayNI

(1 to: 5) do: [:item | ('%1 Bedy te vita.' % { item }) displayNI]

- Smalltalk není jen jazyk (na rozdíl od Javy, C#). Jde o systém s vlastnostmi běžně očekávanými od operačního systému. Jazyk je jeho součástí a v tomto jazyce je celý systém vytvořen. Díky tomu je schopen sám sebe svými vlastními prostředky vyvíjet (inkrementálně, za běhu, bez restartu).



0:35

17 / 31

0:40



PHP

- PHP je scriptovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací (formáty WML, HTML, XHTML).
- PHP je nezávislý na platformě, rozdíly v různých operačních systémech se omezují na několik systémově závislých funkcí a skripty lze většinou mezi operačními systémy přenášet bez úprav.

```
<?php  
print("Vitam te u sebe");  
for($i = 1; $i < 6; $i++)  
{  
    print(i + " Bedy te vita.");  
}  
?>  
]
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Prolog

- Prolog je jazyk pro programování symbolických výpočtů. Jeho název je odvozený ze slov Programming in Logic a vychází z principů matematické logiky. Jeho úspěch byl podnětem pro vznik nové disciplíny matematické informatiky – logického programování, což je perspektivní styl programování na vyšší abstraktní úrovni. Prolog je také strojovým jazykem nejmodernějších počítačů. Má doposud specifické oblasti použití jako je umělá inteligence, znalostní inženýrství.



0:35

19 / 31

0:40



ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZÁBAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- **Základním stavebním kamenem programů v Prologu je výroková logika (logiku máme výrokovou a predikátovou).**

uvod(Sez):- Sez=[1,2,3,4,5,6,7,8,9],tabulka(Sez). % nacteni seznamu cisel do Sez, spusti funkci tabulka a vrati Sez do S
vymen(X,Co,[X|T],[Co|T]). % Vymeni zadane cislo (X) v seznamu(T)
vymen(X,Co,[H|T],[H|T1]):-vymen(X,Co,T,T1). % za znak hrace (Co) - kolecko nebo krizek
vytkni(X,[X|T],T). % vezme cast seznamu a vrati její cast
vytkni(X,[H|T],[H|T1]):-vytkni(X,T,T1).
spoj([],Sez,Sez). % spoji dve rozdelene casti hraci plochy
spoj([H|T],S,[H|W]):-spoj(T,S,W).
tabulka([A1,A2,A3,B1,B2,B3,C1,C2,C3]):- % funkce tabulka prijma 9 cisel v poli



0:35

20 / 31

0:40



ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- **Výroková logika** používaná v prologu je značně jednoduchá. Formuluje věty pomocí výroků a logických spojek (spojuje složitější formule např. do tvaru implikace – podmínky).
- **Predikátová logika** používá také predikáty, funktory a proměnné (umožňuje formulovat vztahy a vlastnosti objektů pomocí relací). Důležitou charakteristikou predikátové logiky jsou kvantifikátory, které je nutno při logickém programování obětovat. Všechny proměnné jsou chápány jako univerzální. Predikátová logika dává schopnost pracovat nejen s elementárními výroky, ale také rozlišit objekty a jejich vztahy



0:35

21 / 31

0:40



ROBOTI

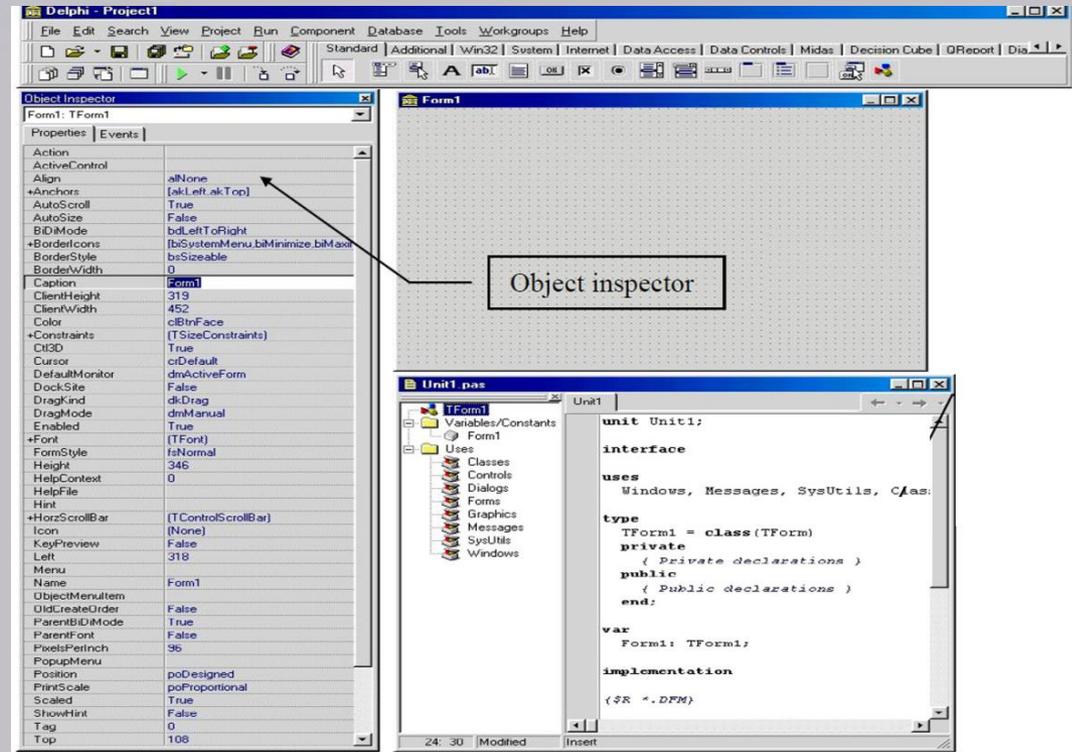
VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Delphi

- Prostředí je založeno na vizuálním principu. Všechno, co bude v běžícím programu vizuálně zobrazeno, programátor během návrhu programu vše vizuálně skládá z předpřipravených částí (komponent).



Object inspector

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Python

- Jde o dynamický interpretovaný jazyk. Bývá dokonce považován za jeden z nejvhodnějších programovacích jazyků pro začátečníky.

```
invoice_creator.py - Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
New Open Back Forward Save Save As Close Undo Redo
Documents
  er75
    mysql.py
    server_axam.py
  internet
    invoice creat...
  mail_to_sms
  imap_tester.py
Projects
124 >> def add_item(self, title, description, cost, tax="21"):
125 >>     #print cost
126 >>     self.items.append({"title": title, "description":description, "cost": cost, "tax":tax})
127 >>     self.data["cost_with_tax"] += cost
128 >>     if tax:
129 >>         self.data["cost_without_tax"] += cost/1.21
130 >>     else:
131 >>         self.data["cost_untaxed"] += cost
132 >>
133 >>
134 >> def replace_by_dict(self, document, dict_):
135 >>     ret = unicode(document)
136 >>     for i in dict_:
137 >>         #print dict_[i]
138 >>         if type(dict_[i]) == types.IntType or type(dict_[i]) == types.FloatType:
139 >>             ret = ret.replace("{"+i+"}",unicode(dict_[i]))
140 >>         else:
141 >>             ret = ret.replace("{"+i+"}",dict_[i])
142 >>     return ret
143 >>
144 >> def render(self, template_file, output_file):
145 >>     f = codecs.open(template_file, mode="r", encoding="utf-8")
146 >>     template = f.read()
147 >>     f.close()
148 >>     f = codecs.open(template_file + ".item", mode="r", encoding="utf-8")
149 >>     item_tpl = f.read()
150 >>     f.close()
151 >>
152 >>     for i in self.items:
```

Line: 234 Col: 5 INS LINE UTF-8 invoice_creator.py
Current Project

Python

- Jde o dynamický interpretovaný jazyk. Bývá dokonce považován za jeden z nejvhodnějších programovacích jazyků pro začátečníky.
- Python je hybridní jazyk (nebo také víceparadigmatický), to znamená, že umožňuje při psaní programů používat nejen objektově orientované paradigma, ale i procedurální a v omezené míře i funkcionální, podle toho, co se pro danou úlohu nejlépe hodí.

```
#!/usr/bin/python  
print "Vitam te u sebe."  
for i in range(4):  
    print i+1, "Bedy te vita."
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- **Jedna z největších výhod Pythonu je podpora velkého množství datových typů. Podporuje seznamy, asociativní pole, posloupnosti a spoustu dalších typů.**

```
124 > def add_item(self, title, description, cost, tax="21%"):
125 >     #print cost
126 >     self.items.append({"title": title, "description":description, "cost": cost, "tax":tax})
127 >     self.data["cost_with_tax"] += cost
128 >     if tax:
129 >         self.data["cost_without_tax"] += cost/1.21
130 >     else:
131 >         self.data["cost_untaxed"] += cost
132 >
133 >
134 > def replace_by_dict(self, document, dict_):
135 >     ret = unicode(document)
136 >     for i in dict_:
137 >         #print dict_[i]
138 >         if type(dict_[i]) == types.IntType or type(dict_[i]) == types.FloatType:
139 >             ret = ret.replace("%+i+",unicode(dict_[i]))
140 >         else:
141 >             ret = ret.replace("%+i+",dict_[i])
142 >     return ret
143 >
144 > def render(self, template_file, output_file):
145 >     f = codecs.open(template_file, mode="r", encoding="utf-8")
146 >     template = f.read()
147 >     f.close()
148 >     f = codecs.open(template_file + ".item", mode="r", encoding="utf-8")
149 >     item_tpl = f.read()
150 >     f.close()
151 >
152 >     for i in self.items:
```

Java

- Jedná se o objektově orientovaný jazyk vycházející z C++.
- Velkou výhodou Javy je také její hardwarová nezávislost, neboť je překládána do speciálního mezikódu (bytecode), který je na konkrétním počítači interpretován, příp. za běhu překládán do nativního kódu.

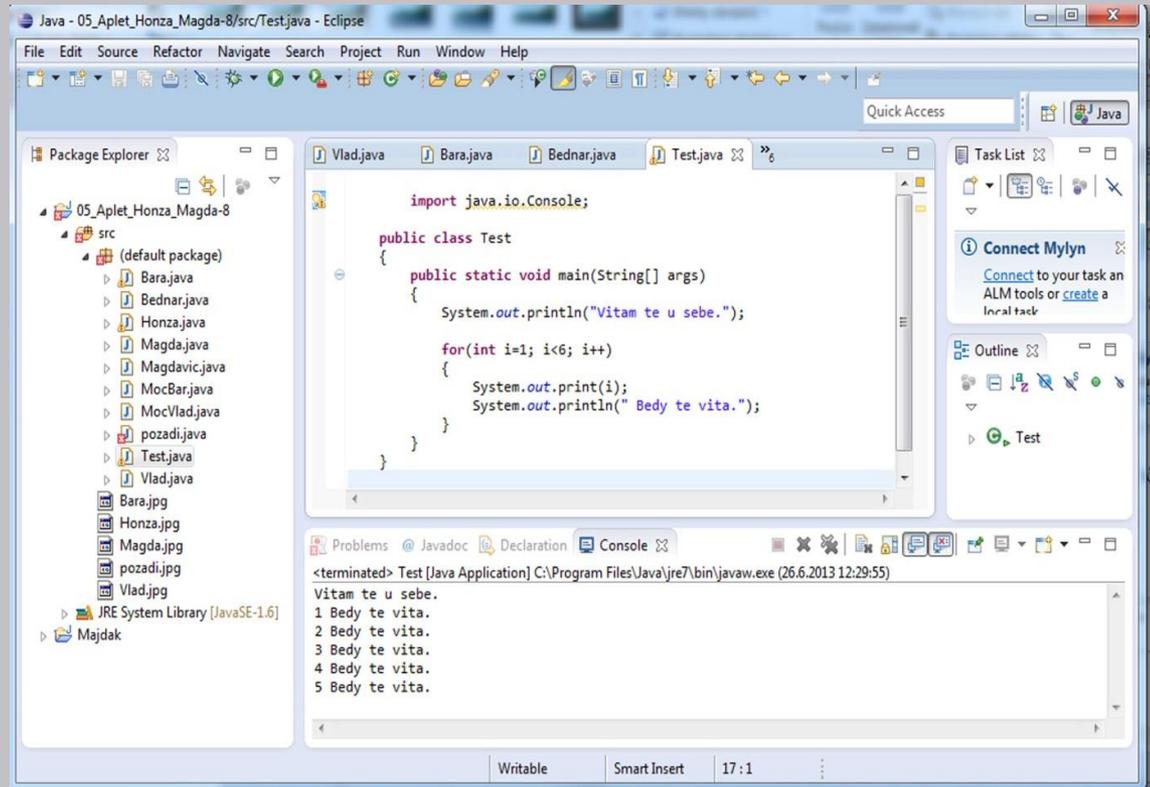
```
import java.io.Console;  
public class Test  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Vitam te u sebe.");  
        for(int i=1; i<6; i++)  
        {  
            System.out.print(i);  
            System.out.println(" Bedy te vita.");  
        }  
    }  
}
```

ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



- **Jáva má množství standardně dodávaných knihoven, se kterou se nemůže srovnávat asi žádný běžně používaný jazyk.**



ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZÁBAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Java Script

- JavaScript je internetový jazyk, doplňující HTML kód o některé zajímavé prvky. Běží na klientské straně - prohlížeč si stáhne zdrojový kód (který se vepisuje přímo do HTML kódu) a teprve potom ho spustí (pokud to umí).
- JavaScript je jazyk objektový. Příčinou vzniku JavaScriptu byl požadavek na zvýšení uživatelského komfortu pro uživatele internetových stránek.



0:35

28 / 31

0:40



ROBOT

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

POUŽITÉ INTERNETOVÉ ZDROJE

Úvod a historie C++. In: Devbook: Programátorská sociální síť [online]. [cit. 2013-06-21]. Dostupné z: <http://www.devbook.cz/kurz-cpp-uvod-do-sveta-cpp>

Učebnici Assembleru: Terminologie. [online]. [cit. 2013-06-21]. Dostupné z: <http://conmet.cz/assembler/uc02.htm>

ASSEMBLER: Asm obecně. [online]. [cit. 2013-06-21]. Dostupné z: <http://k-prog.wz.cz/progjaz/assemb.php>

PYTHON: Úvod k Pythonu. [online]. [cit. 2013-02-21]. Dostupné z: <http://k-prog.wz.cz/python/index.php>

PASCAL: Úvod do Pascalu. [online]. [cit. 2013-06-21]. Dostupné z: <http://k-prog.wz.cz/pascal/index.php>

ŠTEFAN, Radim. Programování: studijní obor: informační technologie ve vzdělávání.

Vyd. 1. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2002, s. 12. Informační technologie ve vzdělávání. ISBN 80-7042-254-8.

C++. In: [online]. [cit. 2013-06-25]. Dostupné z: http://www.bloodshed.net/images/devcpp5_scr.jpg

NAGYOVÁ, Ingrid. Základy programování: vývojové prostředí DELPHI. Vyd. 1. Ostrava: Ostravská univerzita, Pedagogická fakulta, s. 3. Informační technologie ve vzdělávání.



ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

POUŽITÉ INTERNETOVÉ ZDROJE

ŠTEFAN, Radim. Autorské systémy: vývojové prostředí DELPHI : studijní obor: Informační technologie ve vzdělávání. Vyd. 1. Ostrava: Ostravská univerzita, Pedagogická fakulta, 2002, s. 20. Informační technologie ve vzdělávání. ISBN 80- 7042-253-x.

1. díl - Programování v jazyce C - Úvod. In: Devbook: Programátorská sociální síť [online]. [cit. 2013-06-21]. Dostupné z: <http://www.devbook.cz/jazyk-c-uvod-tutorial>

Oracle: Java Embedded Technology Enables End-to-End Solutions. [online]. [cit. 2013-06-21]. Dostupné z: <http://www.arm.com/community/software-enablement/oracle-foundation-page-22516.php>

JAVASCRIPT: Co je JavaScript. [online]. [cit. 2013-06-21]. Dostupné z: <http://kprog.wz.cz/javascr/index.php>



ROBOTI

VE ŠKOLE PRO PRAKTICKOU VÝUKU, MOTIVACI I ZABAVU



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Konec

Děkuji všem přítomným za pozornost.

Autor : Vladislav Bednář
Kontakt : bednar@sse-najizdarne.cz
Vytvořeno : 31. 1. 2014

**Střední škola elektrotechnická, Ostrava,
Na Jízdárně 30, příspěvková organizace**

