



Základní škola sv. Voršily v Olomouci

Aksamitova 6, 772 00 Olomouc

Programovací jazyky

Závěrečná práce

Autor: David Lachman

Třída: IX.

Vedoucí práce: Mgr. Vilém Lukáš

Olomouc 2015

OBSAH

ÚVOD	4
Kapitola I - Pojmy a jejich vysvětlení	5
Kapitola II - Co to je Programovací jazyk?	6
1 Jak se dá definovat programovací jazyk	6
2 Dělení programovacích jazyků	6
KAPITOLA III – Zpracování a přenos informací	7
1 Rozdíl mezi člověkem a počítačem ve zpracování a přenosu informace	7
1.1 Zpracování informací u člověka	7
1.2 Zpracování informací u počítače	7
1.3 Shrnutí rozdílů o zpracování informací	8
2 Binární soustava	8
3 Převody jednotek	9
KAPITOLA IV – Populární jazyky	10
1 Java	10
1.1 Historie	10
1.2 Využití	11
1.3 Syntaxe a vzhled	11
1.4 V čem vyniká?	12
2 C++	13
2.1 Historie	13
2.2 Využití	14
2.3 Syntaxe a vzhled	14
2.4 V čem vyniká?	15
3 COBAL	16

3.1 Historie	16
3.2 Využití	16
3.3 Syntaxe a vzhled	17
3.4 V čem vyniká?	18
ZÁVĚR	19
POUŽITÁ LITERATURA	20
REZUMÉ	21

Úvod

Programovací jazyky. Mnozí již určitě slyšeli tento pojem nebo zaslechli. Ale opravdový význam či definici těchto slov by někomu, kdo o tom ještě nic neví, vysvětlil málokdo. V horším případě, pokud dotyčný věří na pověry, řekl by: „To jsou slova kouzelníků.“ A právě tyto pomluvy se tato práce snaží vyvrátit. Čtyři kapitoly, sto padesát šest řádků a víc jak tisíc pět set šedesát slov to je rozmezí, které se vám bude snažit ukázat, že programovací jazyky nejsou záležitostí kouzelnických praktik, ale programy, vytvořené lidmi pro lidi.

Programovací jazyky jsou velmi rozsáhlé téma, které se dá rozdělit na spousty kapitol, podkapitol atd. jenže můj záměr není z amatéra vytvořit specialistu na programovací jazyky, nýbrž uvedení čtenáře do obrazu, aby již nemusel zaostávat v dnešním světě technologií. Z tohoto důvodu se zde nedočtete o podrobných informacích k danému tématu, ale pouze lehkému nastínění témat, které jsou utvořeny tak, aby to pochopit každý obyčejný člověk bez zbytečného dohledávání dodatečných informací.

Obsah této práce se tedy zabývá velmi stručnou formou každé problematiky, k jejímu pochopení si vystačíte i s uvedenými informacemi v textu, ve kterých se můžete dozvědět:

- orientace ve světě počítačových jazyků (pojmy pro lepší pochopení následujících kapitol)
 - zpracovávání a přenos informací
 - jak se jazyk správně definuje
 - jak vypadají (Java, C++, COBOL) nejpoužívanějšími jazyky a proč zrovna tyto jazyky

KAPITOLA I – Pojmy a jejich vysvětlení

- software – Pod známějším pojmem programy se označuje posloupnost příkazů, podle kterých počítač pracuje. Pro vykonání jakékoli činnosti musí počítač použít příslušný software.
- zdrojový kód - Jedná se o příkazy, které vepisujeme do programovacího jazyka.
- kompilátor – Přeložený význam slova kompilátor je míněn překladač. To znamená, že kompilátor něco překládá, např. zdrojový kód přeloží do bajkódu.
- syntaxe – Syntaxe definuje pravidla pro správný zápis instrukcí. Např. říká kde, patří velké písmeno a kde malé.
- sématika – Sématika definuje funkce každé konstrukce. Např. operátor = provede přiřazení ($x=3$).
- binární soustava – Soustava, jejímž základem je 1 a 0. Tato soustava je základem každé informace v počítači.

KAPITOLA II - Co to je Programovací jazyk?

1 Jak se dá definovat programovací jazyk

Forem vyjádření co je to programovací jazyk je mnoho a často se můžete setkat s trochu jinou formou a to kvůli výkladu, který vám poskytne každý trochu jinak, ale vždy musí mít základní myšlenku stejnou. Já zde poskytnu tu variantu výkladu, která mě přišla nejpochoptelnější.

Programovací jazyky jsou virtuální prostředí, do kterého zapisujeme algoritmy neboli zdrojové kódy. Díky programovacím jazykům můžeme tvořit softwary, jak je známe např. Facebook, Tetris, Microsoft Word, GIMP atd.

2 Dělení programovacích jazyků

Programovací jazyky můžeme dělit spousty způsoby. Nejčastěji se setkáme se způsobem dělení podle účelu, který se dělí dále na tři typy.

- **NÍZKOÚROVŇOVÉ** - Jazyky přizpůsobené pro řízení činnosti počítače na nízké úrovni - to znamená jazyky, které pracují s operačním systémem např. Windows. Tento způsob účelu využívá např. Assembler
- **VYSOKOÚROVŇOVÉ** – Tento typ jazyku se využívá pro tvorbu samotných aplikací, jako jsou např. Microsoft Powerpoint, Microsoft Word atd. K této skupině se přiřazuje např. BASIC, C, COBOL, Visual Basic
- **JAZYKY NA APLIKAČNÍ ÚROVNI** – Tato skupina je určena pro ovládání aplikačních programů, jako je třeba Microsoft Powerpoint, Microsoft Word. Jako jazyky na aplikační úrovni se pokládá VBA Word, VBA Powerpoint. Všechny tyto jazyky jsou stejné, akorát ovládají jiný program, proto se vždy vyskytuje ve jménu VBA.

KAPITOLA III – Zpracování a přenos informací

1 Rozdíl mezi člověkem a počítačem ve zpracování a přenosu informace

Mezi člověkem a počítačem je velký rozdíl pokud jde o zpracování informací. Proto, aby nedošlo, zbytečnému zmatení si uvedeme jednotlivé schémata jak u počítače, tak u člověka.

1.1 Zpracování informací u člověka

Nejprve musí být myšlenka (data), která je prostřednictvím úst vyslovena. Příjemce vyslovenou myšlenku začne zpracovávat a to probíhá ve dvou krocích.

1. vytažení důležitých informací z dat vznikne capta
2. z capty se vytváří informace na základě dosavadních poznatků

Po těchto dvou krocích je již informace ukládána do mozku a připravena k budoucímu využití.

**MYŠLENKA → ÚSTA → MOZEK → CAPTA →
→ INFORMACE → UKLÁDÁNÍ**

1.2 Zpracování informací u počítače

V programovacím jazyku napíšeme zdrojový kód. Ten je následně v kompilátoru přeložen do bajkódu, který je interpretován v JVT do nativního kódu a na konec je nativní kód spustí v procesoru.

**ZDROJOVÝ KÓD → KOMPILÁTOR → BAJKÓD → JVT →
NATIVNÍ KÓD → PROCESOR**

1.3 Shrnutí rozdílů o zpracování informací

Ze schémat o zpracování informací u člověka i počítače vyplívá jeden velmi podstatný rozdíl a to že člověk s informacemi pracuje, přetváří nebo někdy i upravuje v průběhu cyklu. Zatím co když píše programátor na počítači zdrojový kód, nechává počítač každé písmeno stejné od napsání programátorem, až do processoru kde je vykonán příkaz.

2 Binární soustava

Všichni známe desítkovou soustavu, protože ji používáme, ale tu binární, která se stává nedílnou součástí našeho života? Pokud ne tak se seznáme s důležitou součástí šifrování a dešifrování v informací počítači. Binární soustava neboli dvojková. Na které se zakládá vše, co děláte v programovacím jazyce tak v celém počítači.

Při převodu čísla z desítkové soustavy do dvojkové postupujeme takto: číslo vydělíme celočíselně dvěma. Zbytek po celočíselném dělení (má hodnotu 0 nebo 1) představuje jednu číslici ve dvojkovém zápise čísla. Převod do dvojkové soustavy číslo ve dvojkové soustavě Výsledek dělení opět vydělíme celočíselně dvěma a zjistíme zbytek po dělení. Celý postup opakujeme tak dlouho, dokud je výsledek dělení nenulový. Dvojkový zápis čísla je tvořen zbytky nalezenými v předchozích krocích. Tyto zbytky zapíšeme v opačném pořadí, než jsme je získali.

zbytek po dělení		celočíselné dělení
	↓	↓
14 mod 2 = 0	← 0. řád	14 / 2 = 7
7 mod 2 = 1	← 1. řád	7 / 2 = 3
3 mod 2 = 1	← 2. řád	3 / 2 = 1
1 mod 2 = 1	← 3. řád	1 / 2 = 0
$14_{10} = 1110_2$		

Obrázek 1. Z desítkové do binární soustavy

Při převodu čísla z dvojkové soustavy do desítkové postupujeme ve dvojkovém zápise zprava a za každou jedničku na n-tém místě přičteme hodnotu 2^n . Pozice číslujeme od 0.

$$\begin{array}{cccc} 2^3 & 2^2 & 2^1 & 2^0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 1 & 1 \\ \hline & & & 2 \end{array} = (8 + 2 + 1)_{10} = 11_{10}$$

Obrázek 2. Z binární do desítkové soustavy

3 Převody jednotek

Paměť počítače je složena ze základních paměťových buněk, jenž mohou mít buď hodnotu 1 nebo 0. Tato základní buňka se jmenuje bit. Dále 8 bitů je bajt a 4 bajty vytvoří word a další převody jsou následující:

$$\begin{aligned} 1 \text{ KB} &= 1024 \text{ B} \sim 10^3 \text{ B} \\ 1 \text{ MB} &= 1024 \text{ KB} \sim 10^6 \text{ B} \\ 1 \text{ GB} &= 1024 \text{ MB} \sim 10^9 \text{ B} \\ 1 \text{ TB} &= 1024 \text{ GB} \sim 10^{12} \text{ B} \end{aligned}$$

Obrázek 3. Převody jednotek

KAPITOLA IV – Populární jazyky

V této kapitole se pojednává o třech rozšířenějších jazycích. U každého je vždy uvedena krátká historie, využití, jak vypadá syntaxe, vzhled a proč by bylo vhodné si zvolit zrovna tento jazyk.

1 Java



Obrázek 4. Logo Javy

1.1 Historie

Historie tohoto jazyka začíná, počátkem 90. let, kdy byl složen, Green Team společnosti Sun World pod vedením kanadského programátora Jamese Goslinga, Tento tým se po dlouhé dva roky snažil vytvořit program pro domácí spotřebiče. Podle plánu by výsledný výtvar měl zprostředkovávat ovládání spotřební elektroniky na dálku.

Tuto idey se snažili vytvořit modifikací jazyka C++, ale výsledky nebyli příznivé. Po dlouhém bádání Jamese Goslinga navrhl úplně noví jazyk jménem Oak (dub), který se dá pokládat za nástupce C++. Tento nápad již splňoval požadavky tvůrců a byl uveden do praxe.

Název Oak byl později pozměněn, kvůli již existujícímu programovacímu jazyku Oak na už známí celosvětový fenomén Java, který byl poprvé představen na konferenci SunWorld 1995.



Obrázek 5. James Gosling

1.2 Využití

Od vydání verze 1.2, začala mít Java široké využití, které se začalo dělit do tří kategorií Java SE (Java Platform, Standard Edition), Java ME (Java Platform, Micro Edition) a Java EE (Java Platform, Enterprise Edition)

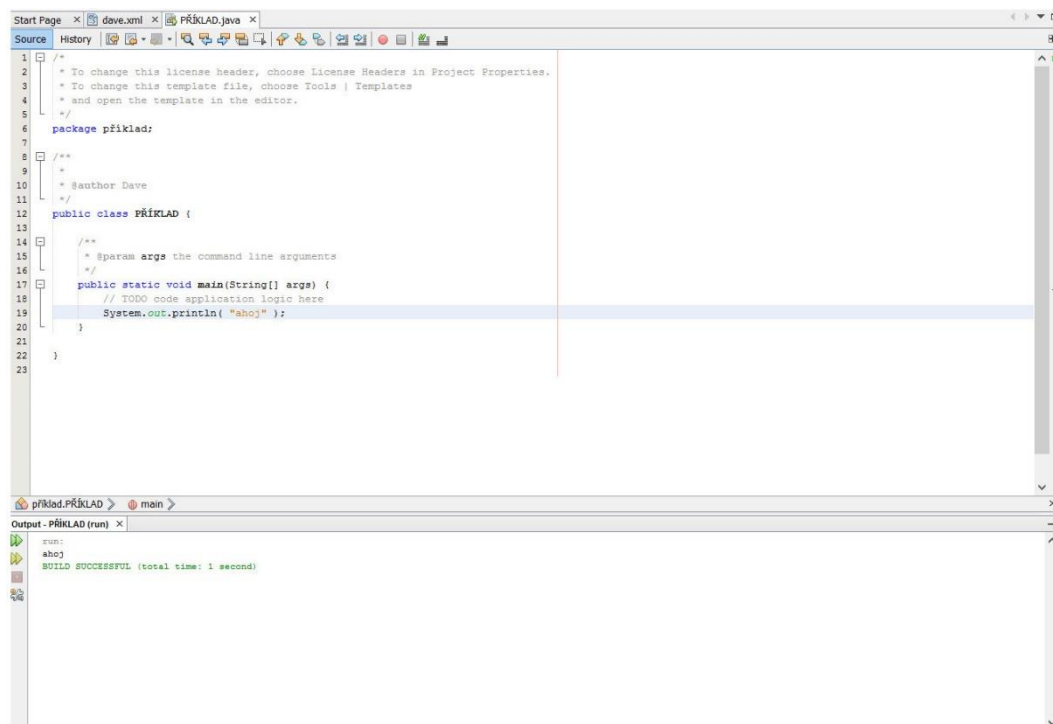
- Java SE – Jedná se o základní platformu Javy sloužící k vytváření programů pro stolní počítače
- Java ME – Podmnožina SE určena k vývoji softwaru pro domácí elektroniku, mobilní telefony atp.
- Java EE – Poslední ze základních platform, využívaná k rozvoji a provozu podnikavých aplikací.

1.3 Syntaxe a vzhled

Syntaxe a vzhled je nedílnou součástí každého jazyku a je nesmírně důležité si zapamatovat, co vše znamená a kde to najdeme. Ať už se jedná o Javu, COBEL nebo C++.

Jednoduchým příkladem syntaxe, vzhledu nebo programování je známí program jménem Hello World!, který se vždy uvádí jako seznamování příklad s každým jazykem. Výsledek syntakticky a sématicky správně napsaného programu Hello World! je okno, ve kterém je napsáno „Hello, world!“ v našem případě „ahoj“.

```
public class První {
    public static void main( String[] args ) {
        Szstem.out.println( "ahoj " );
    }
}
```



Obrázek 6. Syntaxe a vzhled v Javě

1.4 V čem vyniká?

Říci že Java je celosvětový fenomén je sice velmi přitažlivé, ale nemněla by tato slova být rozhodujícím faktorem pro výběr jakéhokoli jazyka. A to proto, že vždy je příjemnější vědět přesné možnosti než zbytečně poté plakat nad rozlitym mlékem.

Zde se nacházejí důvody, které by vás měli přesvědčit nebo vám pomoci pro zvolení jazyka Java.

- Java patří mezi jazyky syntakticky jednodušší.
- Dá se použít na mnoha operačních systémech. To znamená, že můžete své projekty otvírat jak v operačním systému Microsoft Windows, tak i v Linuxu a dalších.
- Vhodná i pro začátečníky, lepe se zde hledají chyby.
- Je na dobré výkonnostní úrovni a to ne jen díky kompilátoru JIT
- Použitelná na všechny projekty (již zmiňované Java SE, ME, EE).

2 C++



Obrázek 7. Logo C++

2.1 Historie

Pro pochopení co je C++, potřebujeme znát jeho základ a to základy vzniku programovacího jazyka C. Kořeny jazyka C začínají, někde v šedesátých letech, kdy se začal vyvíjet program, který měl zásadní vliv na to, jak se v budoucnu bude programování ubíhat.

První krok pro vytvoření zlomového jazyka byl vznik nového operačního systému UNIX (1968 v Bellových laboratořích v Murray Hill, New Jersey), který je s C silně historicky spjat. Brzy po vzniku UNIXu začal Ken Thompson vytvářet pro již zmiňovaný operační systém nový jazyk pod názvem B.

Teď se vraťme o kousek zpátky a to do roku 1963 na univerzitu Cambridge a University of London kde byl vytvořen jazyk CPL. Tento jazyk byl, ale příliš složitý na pochopení a proto byl o čtyři roky později tedy roku 1967 upraven Martinem Richardem na jazyk BCPL (Basic Combined Programming Language).

Poslední důležitou událostí pro vznik C byl nový typ počítače PDP-11 pro který se hledal vhodný jazyk. Na tento post kandidoval jazyk B, jenže tento návrh byl zamítnut, kvůli pomalosti B a nevhodnosti pro psaní operačního systému. Toto byl poslední podnět pro vytvoření dalšího nástupce B a tím se stal nový jazyk C roku 1972, za jehož autora se poklád Dennis M. Ritchie. Tento, nový jazyk se vracel k některým obecným principům CPL, jenž BCPL a B postrádali.

A z jazyka C se později díky Bjarna Stroustrupa z Bellových laboratořích vyvinula nová verze, která poprvé spatřila svět roku 1983 pod názvem C++.



Obrázek 8. Dennis M. Ritchie



Obrázek 9. Bjarn Stroustrup

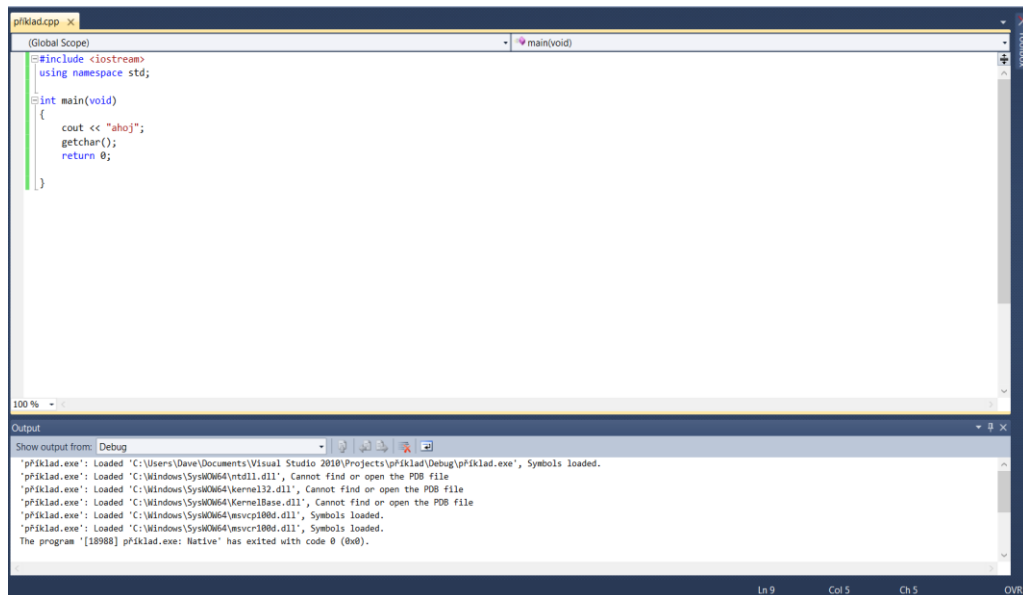
2.2 Využití

C++ pochází z Bellových laboratořích, které vyvinuly jazyk velmi vhodný pro ovládání počítače. Ve srovnání s Javou bychom řekli, že má podobný účel jako Java SE.

2.3 Syntaxe a vzhled

```
#include <iostream>
using namespace std;

int main ()
{
    cout << "Hello World!";
    getchar();
    return 0;
}
```



Obrázek 10. Syntaxe a vzhled v C++

Nyní už známe provedení stejného programu ve dvou jazycích. Jenže na první pohled vypadá týž program jinak než ve druhém a trochu také je, ale pouze v zápise nikoli ve výsledném produktu. Tento rozdíl je dán pouze syntaktikou, jež se liší v každém jazyku.

2.4 V čem vyniká?

C++ patří také mezi špičky programovacích jazyků jako Java, ale mimo jiné se může pyšnit vysokou rychlostí zpracováním dat, ve které rozhodně za Javou nezaostává. Dalším bodem proč zrovna C++ je také jednoduchá syntaktika ve srovnání s jinými jazyky a na konec podle mého názoru je přehlednější či lépe utvořená orientačně.

3 COBEL



Obrázek 11. Logo COBOL

3.1 Historie

Počátky programovacího jazyka COBOL (COmmon Business Oriented Language) byly uvedeny do pohybu již na konci padesátých let minulého století, jedná se tedy o jeden z nejstarších dodnes používaných programovacích jazyků. V roce 1959 se na konferenci pořádané v USA sešli zástupci firem a některých vládních i nevládních organizací, aby se dohodli na vytvoření vysokoúrovňového programovacího jazyka vhodného pro zpracování dat i pro řízení různých procesů ve firmách. V průběhu této konference byly vytvořeny zvláštní komise vedené výkonným výborem CODASYL (Conference on Data Systems Languages). Výsledkem práce těchto komisí byla publikace Initial Specification for a Common Business Oriented Language (COBOL) for Programming Electronic Digital Computer, která byla vydána v roce 1960. Tato publikace se později stala mezi odborníky známá pod zkratkou COBOL-60, protože na jejíž základě vznikly i první implementace jazyka COBOL.

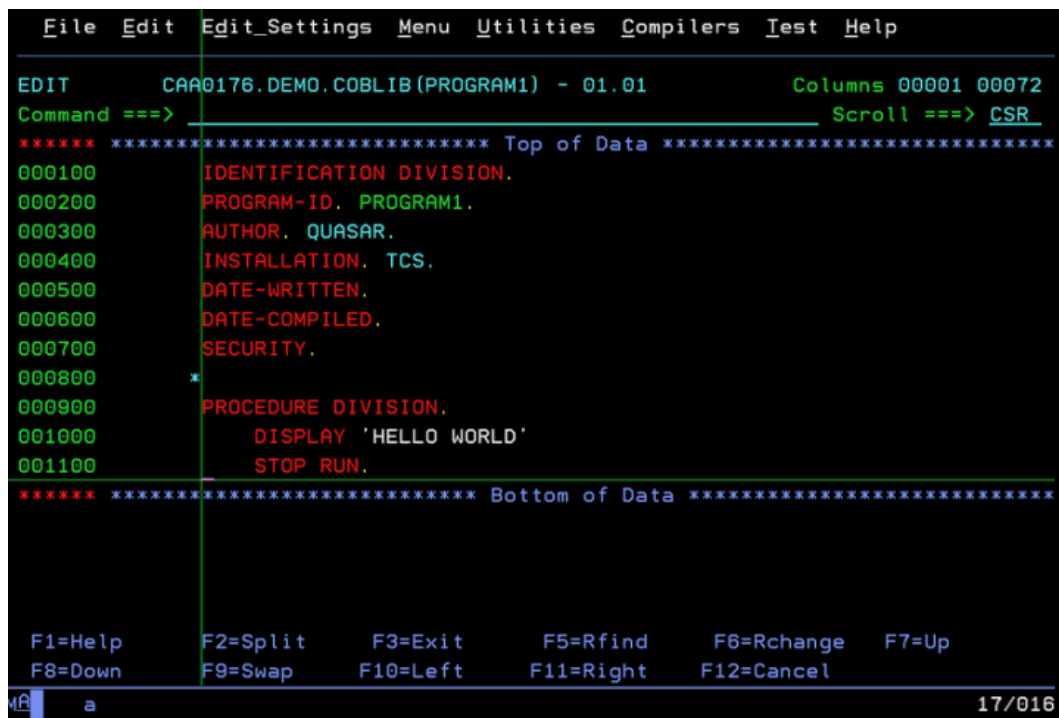
3.2 Využití

Původní myšlenkou výboru CODASYL bylo vytvořit jazyk na takové úrovni, že by ho dokázal ovládat i člověk, který nemá s programováním žádné

zkušenosti. Tento vztyčený cíl byl splněn a dnešní COBOL je využíván k různým finančním nebo archivovacím účelům normálního člověka.

3.3 Syntaxe a vzhled

```
IDENTIFICATION DIVISION.  
  
PROGRAM-ID. HELLO-WORLD.  
  
ENVIRONMENT DIVISION.  
  
DATA DIVISION. PROCEDURE DIVISION.  
  
DISPLAY "HELLO, WORLD".  
  
STOP RUN.
```



```
File Edit Edit_Settings Menu Utilities Compilers Test Help  
EDIT CAA0176.DEMO.COBLIB (PROGRAM1) - 01.01 Columns 00001 00072  
Command ==> Scroll ==> CSR  
***** Top of Data *****  
000100 IDENTIFICATION DIVISION.  
000200 PROGRAM-ID. PROGRAM1.  
000300 AUTHOR. QUASAR.  
000400 INSTALLATION. TCS.  
000500 DATE-WRITTEN.  
000600 DATE-COMPILED.  
000700 SECURITY.  
000800 *  
000900 PROCEDURE DIVISION.  
001000 DISPLAY 'HELLO WORLD'  
001100 STOP RUN.  
***** Bottom of Data *****  
F1=Help F2=Split F3=Exit F5=Rfind F6=Rchange F7=Up  
F8=Down F9=Swap F10=Left F11=Right F12=Cancel  
17/016
```

Obrázek 12. Syntaxe a vzhled v COBOLu

Syntaxe v COBOLu je tedy udělána tak, aby byla co nejjednodušší pro pochopení. To ovšem má své následky, rychlost kódování a dekodování zdrojového kódu je na rozdíl od Javy a C++ znatelně menší.

3.4 V čem vyniká?

Na rozdíl od Javy a C++ je COBOL soustředěn pouze na jednu věc a to pro firemní účely. To ovšem z COBOLu dělá specifickou skupinu, ve které je téměř osamocený. Proto není moc možností jaký jazyk zvolit pro tento typ práce a to je chybou. Kvůli své samotě nemá žádný důvod se rozvíjet. Popravdě to je jediný důvod, proč je COBOL špičkou víc než padesát let.

ZÁVĚR

Cílem této práce bylo přiblížení programovacích jazyků čtenářům, kteří chtějí získat základy a vytvoření si vlastního názoru na jazyky. Tento cíl byl naplňován pomocí kapitol o tom, co je programovací jazyk samotný, jak vypadá kódování informací v počítači a představením několika používanějších programovacích jazyků. Tyto kapitoly se dále rozvíjeli a bylo v nich možné zjistit, že programovací jazyky jsou jakési virtuální prostředí dělicí se podle účelu využití. Dále byla objasněna otázka, jak probíhá proces přenosu dat mezi jednotlivými částmi počítače. A poslední možností jak se dozvědět něco o tomto tématu byla prezentace historie, využití, syntaxe a vzhledu jazyků Java, C++ a COBOL.

Doufám, že vám tato práce rozšířila vědění o užitečné informace z okruhu programovacích jazyků.

POUŽITÁ LITERATURA

1. ROMAN, Steven. *Programování makra ve Wordu*. 1. vyd. Praha: Computer Press, 2000. 5 s., 7 s., 49 s.,
2. TIŠNOVSKÝ, Pavel. *Programování mainframů: COBOL*. Citováno 15. 12. 2009. Dostupné z webu: <http://www.root.cz/clanky/programovani-mainfram-cobol/>
3. NEZNÁMO, neznámo. *Principy programování*. Citováno 6. 9. 2009. Dostupné z webu: <http://principyprogramovani.blog.zive.cz/2009/09/style-programovani/>
4. NEZNÁMO, neznámo. *First C++ program, hello world*. Citováno ?. ?. ????. Dostupné z webu: <http://www.codingunit.com/cplusplus-tutorial-first-cplusplus-program-hello-world>
5. NEZNÁMO, neznámo. *Úvod do jazyka C++*. Citováno ?. ?. ????. Dostupné z webu: http://www.ncbr.muni.cz/~martinp/C3220/PCChem_EX1.pdf
6. HULÁK, Radek. *Nejlepší programovací jazyk je C++*. Citováno 5. 8. 2005. Dostupné z webu: <http://myego.cz/item/nejlepsi-programovaci-jazyk-je-c>
7. ČERNOHORSKÝ, Pavel. *Historie a vývoj jazyka C (od C až po C#)*. Citováno ?. ?. 2003. Dostupné na webu: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xcernoh1.htm>
8. NEZNÁMO, neznámo. *Java - Basic Syntax*. Citováno ?. ?. ????. Dostupné na webu: http://www.tutorialspoint.com/java/java_basic_syntax.htm
9. NOVOTNÝ, Luděk. *Historie a vývoj jazyka Java*. Citováno ?. ?. 2003. Dostupné na webu: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xnovotn8.htm>
10. ŽABKA, Michael. *Počítačové zpracování informace*. Citováno ?. ?. 2011/12. Dostupné na webu: http://zabka.avantcore.cz/userfiles/downloads/michael_zabka_pocitacove_zpracovani_informace.pdf
11. NEZNÁMO, neznámo. *PROGRAMOVACÍ JAZYKY*. Citováno ?. ?. ????. Dostupné na webu: <http://k-prog.wz.cz/progjaz/>
12. TRONÍČEK, Zdeněk. *Učebnice jazyku Java*. 1. vyd. Praha: Česká technika, 2007. 2 s., 3 s., 4 s., 6 s. a 7 s.

REZUMÉ

This work is about the programming languages. The intention is to attract readers with basic information that is easily understandable. This means that this work has a purpose programming languages closer to people with not foundations. In selected topics is written on the definition of programming languages, encoding information and selected languages for example Java, C ++ and COBAL.

I hope that this work will something give.