

IB001 - Úvod do programování

RNDr. Jaroslav PELIKÁN, Ph.D.

katedra počítačových systémů a komunikací
Fakulta informatiky Masarykovy univerzity
Botanická 68a, 602 00 BRNO

kanc.: B314, ☎ : +420 - 549 495 751

E-mail: pelikan@fi.muni.cz

<http://www.fi.muni.cz/usr/pelikan>

Organizace předmětu

- 1 přednáška:
 - středa 16.00 - 17.50
- 13 cvičebních skupin:
 - 11 pro studenty Fakulty informatiky (případně jiných fakult)
 - 2 pro studenty Fakulty ekonomicko-správní
- *Zakončení předmětu:*
 - zkouška
 - zápočet

Požadavky ke zkoušce (1)

- Zápočet:
 - „úspěšné“ napsání průběžné písemky
 - splnění všech domácích úkolů (v termínu)
 - maximálně dvě neomluvené neúčasti na cvičení
- Zápočet musí být zapsán na ISu nejpozději 12 hodin před začátkem zkoušky
- Zkouška:
 - teoretická část – písemnou formou
 - praktická část – tvorba programu u počítače

Požadavky ke zkoušce (2)

- **Teoretická část zkoušky:**
 - max. 30 bodů
 - na úspěšné napsání je zapotřebí min. 15 bodů
 - studenti, kteří úspěšně nenapíší teoretickou část, nemohou vykonávat část praktickou
 - bodové hodnocení z teoretické části se započítává do celkového hodnocení zkoušky
- **Praktická část zkoušky:**
 - max. 20 bodů
 - na úspěšné napsání je zapotřebí min. 10 bodů
 - student musí nejprve úspěšně napsat teoretickou část

Požadavky ke zkoušce (3)

- Celkové hodnocení zkoušky:
 - **A**: 45 – 50 bodů
 - **B**: 40 – 44 bodů
 - **C**: 35 – 39 bodů
 - **D**: 30 – 34 bodů
 - **E**: 25 – 29 bodů
 - **F**: 0 – 24 bodů

Literatura (1)

- Pelikán, Jaroslav: *Prezentace k přednáškám z předmětu IB001 Úvod do programování*
<http://www.fi.muni.cz/usr/pelikan>
- *Borland Pascal 7.0 manuál*, Borland International, 1992
- Hála, Tomáš: *Učebnice Pascalu*, Computer Press 2002
- Drózd, Januš - Kryl, Rudolf: *Začínáme s programováním*, Praha: Grada 1992

Literatura (2)

- Wirth, Niklaus: *Algoritmy a štruktúry údajov: Algorithm + Data Structures = Programs*, Bratislava Alfa, 1989
- Teixeira, S. - Pacheco, X.: *Mistrovství v Delphi 6*, Computer Press 2002
- Lischner, R.: *Delphi v kostce*, Computer Press 2000

Programovací jazyky

- Strojový kód
- Jazyk symbolických instrukcí - Assembler
 - vědeckotechnické výpočty
- FORTRAN, ALGOL, COBOL
 - vědeckotechnické výpočty
 - hromadné zpracování dat
- SIMULA, LISP, SIMSCRIPT
- PL/1, ADA, Prolog, C/C++, C#, Pascal, Java

Překladač - Compiler

- Program provádějící transformaci určitého textu (alfanumericky kódované informace - např. program v Pascalu) na posloupnost bytů reprezentující program ve strojovém kódu

„Životopis“ programu

- Zdrojový kód (např. Pascal)
- Makroexpanze
- Překlad (compilation)
 - lexikální, syntaktická a sémantická analýza
- Sestavení (linking)
- Spuštění programu

Programovací jazyk Pascal (1)

- prof. Niklaus Wirth, Švýcarsko 1971 - 1973
- Základní cíle:
 - vytvořit jazyk vhodný pro výuku programování
 - definovat jazyk, jehož implementace na současných počítačích vedou k efektivním a spolehlivým programům

Programovací jazyk Pascal (2)

- Hlavní výhody:
 - standardizace řídicích struktur, které vedou k vytváření dobře strukturovaných a čitelných programů
 - bohatost datových typů
 - relativně snadná přenositelnost na různé typy počítačů

Programovací jazyk Pascal (3)

- 1980 - zveřejněn návrh normy ISO jazyka Pascal
- Borland:
 - Turbo (Borland) Pascal 7.0 (pro MS-DOS)
 - Turbo Pascal for Windows
 - Borland Delphi 7.0 (pro Windows 9x, Me, NT, 2000, XP)

Základní pojmy (1)

- Identifikátor:
 - konečná posloupnost písmen a číslíc začínající písmenem. Překladač rozlišuje vždy jen určitý počet znaků
 - nesmí obsahovat mezeru
 - název volíme mnemonicky
 - např.: teplota, tlak, x_1 , x_2 , ...

Základní pojmy (2)

- Klíčová slova:
 - slova mající v programovacím jazyce svůj speciální význam
 - identifikátor nesmí být pojmenovaný stejně jako klíčové slovo
 - neobsahují mezeru
 - např: program, begin, end, ...

Poznámka

- Dva sousedící identifikátory, klíčová slova nebo čísla musí být od sebe oddělena alespoň jednou mezerou nebo jiným oddělovačem
- Přebytečné mezery a oddělovače řádků jsou překladačem ignorovány
- Pascal nerozlišuje velká a malá písmena (case insensitive)

Struktura programu v jazyce Pascal

program identifikátor;

uses seznam knihovných modulů;

label deklarace návěští;

const definice konstant;

type definice datových typů;

var deklarace proměnných;

deklarace procedur a funkcí;

begin

end.

hlavička

část deklarací
a definic

příkazová
část

Příkazy výstupu (1)

- Příkaz **Write**:
 - **obecný tvar**: `Write (p1, p2, ..., pn)`
 - **funkce**: zobrazí hodnoty parametrů p_1, p_2, \dots, p_n na standardním výstupu (nejčastěji obrazovka monitoru)
- Příkaz **WriteLn**:
 - **obecný tvar**: `WriteLn (p1, p2, ..., pn)`
 - **funkce**: zobrazí hodnoty parametrů p_1, p_2, \dots, p_n na standardním výstupu a následně provede přechod na nový řádek (odřádkování)

Příkazy výstupu (2)

- Poznámky:
 - parametry p_1, p_2, \dots, p_n mohou být:
 - **výrazy** (např. aritmetické nebo logické):
 - příkazy Write a WriteLn zobrazí jejich výsledné hodnoty
 - **proměnné**:
 - příkazy Write a WriteLn zobrazí hodnoty těchto proměnných
 - **řetězce** - posloupnosti znaků uzavřené v apostrofech:
 - příkazy Write a WriteLn provedou jejich zobrazení (bez jakékoliv změny)
 - příkaz WriteLn může být použit i bez parametrů, což způsobí pouze odřádkování

Úprava programu

- Klíčová slova begin a end psát pod sebe
- Příkazy mezi begin a end odsadit o dvě až tři mezery
- Středník oddělující dva příkazy psát bezprostředně za první z nich
- Záhloví programu opatřit komentářem, který obsahuje zadání problému jehož řešení program představuje
- Komentáře - poznámky mezi znaky (*, *) nebo {, }, které jsou překladačem ignorovány

Konstanty a výrazy

- Konstanty:
 - pevně dané hodnoty určitého typu (např. číslo)
 - čísla v Pascalu
 - celá (integer): 264, -1, 8765
 - reálná (real): 3.1415926, -427.812, 1.0, 4E-5, 1.2E3
- Pojmenované konstanty:
 - **const** identifikátor=konstanta;
 - např: `const PI=3.14159; HorniMez=40;`
- Výrazy:
 - tvořeny pomocí +, -, *, /, (,), ...
 - např: $8+4.5/2.0*2$

Formátovaný výstup

- Používán nejčastěji v souvislosti s parametry, jejichž výsledkem je číselná hodnota:
 - formátování celočíselných parametrů:
 - obecný tvar: $X:N$
 - funkce:
 - způsobí, že číslo dané parametrem X je zobrazeno na N míst
 - formátování reálných parametrů:
 - obecný tvar: $X:N:D$
 - funkce:
 - způsobí, že číslo dané parametrem X je zobrazeno celkem na N míst, z nichž D míst je vymezeno pro desetinnou část

Proměnné

- Proměnná:
 - identifikátor, který může být v různých okamžicích realizace programu spojen s různými hodnotami (téhož typu).
 - deklarace:
 - **var** $id_{11}, id_{12}, \dots, id_{1n}$: datový typ₁;
 $id_{21}, id_{22}, \dots, id_{2m}$: datový typ₂;
 - např: var i, j, k: integer;
x, y: real;
- Poznámka:
 - konstanta ne; má pevně danou hodnotu

Příkazy vstupu

- Příkaz **Read**:
 - **obecný tvar**: `Read (p1, p2, ..., pn)`
 - **funkce**: načte hodnoty parametrů p_1, p_2, \dots, p_n ze standardního vstupu (nejčastěji klávesnice)
- Příkaz **ReadLn**:
 - **obecný tvar**: `ReadLn (p1, p2, ..., pn)`
 - **funkce**: načte hodnoty parametrů p_1, p_2, \dots, p_n ze standardního vstupu a následně provede přechod na nový řádek
- Pozn.: V roli parametrů p_1, p_2, \dots, p_n mohou vystupovat pouze proměnné