

Programování v jazyku Pascal

Dvousemestrální kurs *Programování v jazyku Pascal* je formálně dělen na dva vyučovací předměty, jejichž názvy jsou navzájem rozlišeny římskými čísly I a II. Ve školním roce 2005/2006 je vypsán pro následující bakalářské studijní obory (viz tabulka níže). Probíhá formou přednášek a cvičení v zimním (I) a letním (II) semestru školního roku při časové dotaci 2 + 2 vyučovací hodiny týdně.

Studijní obor	Druh studia	Oficiální název předmětu	Zkratka	Status
Finanční matematika	prezenční	Programování v jazyku Pascal I	PGP1	povinný
		Programování v jazyku Pascal II	PGP2	výběrový
	kombinované	Programování v jazyku Pascal I	PGPK1	povinný
		Programování v jazyku Pascal II	PGP2	výběrový
Měřicí a výpočetní technika	prezenční	Programování I	PGP1	povinný
		Programování II	PGP2	povinný
	kombinované	Programování I	PRGD1	povinný
		Programování II	PRGD2	povinný

VÝCHODISKA A CÍLE

Vzhledem k tomu, že na středních školách nepatří výuka programování obecně mezi předměty povinné, nejsou u posluchačů předpokládány žádné speciální vstupní znalosti, jen základní zkušenosti s prací na počítačích třídy PC.

Obsahem předmětu je výuka strukturovaného programování na bázi jazyka Pascal. Tento jazyk byl zvolen proto, že umožňuje tvorbu programů efektivních a přitom snadno modifikovatelných a maximálně srozumitelných. Zásady strukturovaného programování a z nich odvozené datové a příkazové struktury jsou ovšem základem i dalších programovacích jazyků.

Požadované cílové znalosti a dovednosti absolventa tohoto předmětu jsou charakterizovány následujícím výčtem. Absolvent musí dokázat:

- ☞ provést počáteční rozbor úlohy, přesně specifikovat požadované vstupy, omezující předpoklady a výstupy programu, vhodně zvolit vnější a vnitřní reprezentaci dat;
- ☞ formulovat algoritmus činnosti programu, tj. výpočtu dat výstupních z dat vstupních, s využitím metody postupného rozkladu komplexních (složitých) problémů „shora dolů“ na několik problémů speciálních (jednodušších);
- ☞ navrhnout vhodné příkazové struktury pro realizaci algoritmu;
- ☞ podle navrženého algoritmu napsat zdrojový text programu v jazyku Pascal;
- ☞ přeložit, odladit a otestovat program – opravit syntaktické i sémantické chyby a ošetřit činnost programu vůči možným variantám vstupních dat;
- ☞ opatřit program základní dokumentací – v dostatečné míře okomentovat zdrojový text a vytvořit uživatelsky zaměřený popis programu.

Požadované schopnosti student nemůže získat pouhou účastí na přednáškách a cvičeních. Bezpodmínečně nutné je studium alespoň základní literatury a především praktické procvičování teoretických vědomostí na počítači.

TÉMATA PŘEDNÁŠEK A CVIČENÍ V ZIMNÍM SEMESTRU (I)

1. Obecné pojmy: program, programovací jazyk, zdrojový a cílový soubor. Mechanismus překladu, interpretace a kompilace. Charakteristika jazyka Pascal. Struktura zdrojového textu programu. Rozložení programu v operační paměti, datový a kódový segment, omezení. Datové objekty – konstanty a proměnné, jejich typ, deklarace a přiřazení hodnoty. Standardní vstup a výstup dat.

Cvičení: Vstupní informace o práci na počítačových učebnách. Procedura přihlášení, správa účtu, základní operace se soubory a složkami. Základy ovládní integrovaného vývojového prostředí *Borland Pascal 7.0*: operace se soubory (založení, uložení, otevření, uložení pod novým jménem, zavření, přepínání mezi okny); editování textu v okně.

2. Lexikální elementy: speciální symboly, rezervovaná slova, identifikátory, návěští, číselné a řetězcové literály. Oddělovače lexikálních elementů, komentáře. Deklarace konstant a proměnných, konstantní výrazy, proměnné s předdefinovanou hodnotou neboli typové konstanty. Typy celočíselné, reálné, typ znakový a typ logických hodnot.

Cvičení: Ovládní integrovaného prostředí: zápis zdrojového textu programu, překlad a odstranění syntaktických chyb, spuštění programu a ukázka běhové chyby, použití nápovědy. Cvičné programy: načtení znaku a čísla ze vstupu do proměnné, oddělování čísel na vstupu, výpis hodnoty řetězcového a číselného literálu, proměnné a výrazu. Chování proměnné s nedefinovanou hodnotou.

3. Ordinální typy, jejich společné vlastnosti a prostředky. Jednoduché typy, uspořádání. Výrazy, priorita operátorů, typ (hodnoty) výrazu, přetečení. Problémy kompatibility a konverze hodnot mezi celočíselnými typy a mezi celočíselnými a reálnými typy. Výstup dat na obrazovku a jeho formátování. Vstup dat z klávesnice, stavové funkce. Příkazy jazyka Pascal, jejich klasifikace na jednoduché a strukturované. Jednoduché příkazy: prázdný příkaz, přiřazení, volání procedury, skok.

Cvičení: Formátování výstupu. Demonstrace přetečení u ordinálních typů při zapnuté a vypnuté kontrole rozsahu. Ukázka nekonečného cyklu a přerušení běhu programu při vstupu dat a mimo něj. Krokování běhu programu a jeho ukončení.

4. Strukturované příkazy. Sekvence: složený příkaz, uplatnění v příkazových strukturách. Větvení: úplný a neúplný podmíněný příkaz (*if*), příkaz vícenásobného větvení (*case*). Vliv alternativ vyhodnocování logických výrazů na průběh vyhodnocení podmínky. Cykly: s podmínkou na začátku (*while*), s podmínkou na konci (*repeat*) a s pevným počtem průchodů (*for*). Mechanismus nastavení mezí a aktualizace hodnoty řídicí proměnné cyklu *for*, její využití v těle cyklu.

Cvičení: Větvení, prázdný a složený příkaz ve větvi, nadbytečný středník před větví *else*. Cykly. Sledování a úprava hodnoty proměnné a výrazu v průběhu krokování programu, využívání bodů přerušení.

5. Podprogramy. Účel, souvislost s rozkladem úlohy na části, vliv na rychlost výpočtu a na paměťovou náročnost. Deklarace a volání procedury, model blízkého (*near*) a vzdáleného (*far*) volání podprogramu. Komunikace prostřednictvím globálních proměnných resp. parametrů, bezpečnost a variabilita. Seznam formálních a skutečných parametrů, volání hodnotou resp. odkazem, kompatibilita.

Cvičení: Procedury bez parametrů a s parametry, model volání vstupních a výstupních parametrů procedur. Krokování procedur.

6. Funkce. Účel a použití, typ výsledku, omezení. Deklarace a volání funkce, formální zápis předání výsledku a jeho mechanismus. Bloková struktura programu, hierarchie bloků. Globální, lokální a ne-lokální identifikátory, rozsah platnosti, stínění. Vytvoření lokálních proměnných na zásobníku při aktivaci a jejich odstranění při deaktivaci bloku. Lokální typové konstanty, umístění v operační paměti, mechanismus inicializace a důsledky.

Cvičení: Jednoduché programy s procedurami a funkcemi, volání funkcí ve výrazech. Vyvážení vedlejších výsledků výpočtu funkce přes parametry.

7. Rekurzivní podprogramy. Rekurze přímá a vzájemná, dopředná a dodatečná deklarace podprogramu, direktiva *forward*. Mechanismus obnovy lokálních proměnných. Srovnání rekurzivních a iteračních algoritmů podle rozsahu zdrojového textu a kódu, rychlosti výpočtu, paměťových nároků, srozumitelnosti a přehlednosti.

Cvičení: Jednoduché příklady rekurzivních funkcí s lineárně a nelineárně rostoucím počtem volání. Úloha o hanojských věžích jako příklad vhodného použití rekurze.

8. Uživatelské typy. Implementované třídy typů, klasifikace. Popis a identifikátor typu. Typy pojmenované a nepojmenované, synonyma, vlivy na kompatibilitu. Výčtové typy a intervaly. Vlastnosti intervalů, vliv přetečení na výsledek výpočtu.

Cvičení: Realizace vstupně-výstupních konverzí pro výčtové typy, ukázka konverze přetypováním. Využití společných prostředků ordinálních typů pro proměnné a hodnoty výčtových typů. Sledování hodnot výčtových typů během krokování programu. Demonstrace rovnocennosti intervalů a jejich hostitelských typů pro deklarace skalárních proměnných při vypnuté kontrole rozsahu.

9. Typy množina. Popis množinového typu, požadované vlastnosti bazového typu množiny. Konstruktor množiny, prázdná množina, využití intervalů. Vnitřní tvar množinových hodnot, deklarovaný a skutečný rozsah hodnot bazového typu množiny. Množinové operace a relační operace s množinami.

Cvičení: Využití testu na přítomnost prvku v množině místo složené podmínky (například kontrola přípustnosti vstupních dat). Využití množiny ke strádání hodnot bazového typu (generování seznamu znaků vstupního textu).

10. Typy pole. Popis pole, požadavky na typ indexu, počet prvků, využití intervalů. Použitá a deklarovaná délka pole, paměťové nároky. Přístup k prvkům. Vícerozměrná pole, varianty jejich popisu a zápisu přístupu k prvkům. Konstantní pole. Řetězcové typy. Vlastnosti, použití, operace, standardní podprogramy. Deklarovaná a aktuální délka, vnitřní struktura a přístup ke znakům řetězce. Konstantní řetězce. Kompatibilita a typová kontrola řetězců.

Cvičení: Využití pole k uchování seznamu prvků stejného typu v definovaném pořadí, nalezení minima a maxima, vzestupné či sestupné uspořádání prvků pole. Sledování hodnot polí, řetězců a jejich prvků během krokování programu.

11. Typy záznam. Možnost a potřeba soustředění komponent různých typů do společného datového objektu. Prostý přístup ke zvolené položce záznamu aplikací kvalifikátoru a hromadný přístup k položkám příkazem with, problematika stínění identifikátorů. Variantní záznamy. Paměťová úspora, možnost interpretace téhož úseku paměti různými způsoby v jednotlivých variantách. Konstantní záznamy.

Cvičení: Práce s poli záznamů (kartotékami) – vyhledávání prvků podle zvoleného kritéria, uspořádání pole podle hodnot zvolené položky záznamů. Sledování hodnot záznamů a jejich položek během krokování programu.

12. Typy soubor. Komunikace programu s okolím – vstup a výstup dat. Abstrakce souboru v Pascalu, klasifikace souborů podle vnitřní organizace na textové a binární, binárních souborů podle typu komponent na typové a netypové. Popis typu soubor, deklarace ovládací proměnné souboru. Připojení souboru k ovládací proměnné, jeho otevření, zpracování komponent a uzavření. Pořadí operací při zpracování souboru, použití jediné proměnné pro postupné ovládání několika souborů.

Cvičení: Pokračování v práci s poli záznamů.

13. Textové soubory. Organizace a struktura, sekvenční přístup, režimy otevření. Procedury čtení a zápisu, chybové události, testování stavů konec souboru a konec řádky souboru. Textová zařízení, problematika interaktivního vstupu textu z klávesnice. Předdeklarované proměnné *Input* a *Output*, mechanismus automatické obsluhy standardního vstupu a výstupu.

Cvičení: Vstupně-výstupní operace s textovými soubory, demonstrace přesměrování standardního vstupu a výstupu operačního systému.

14. Binární soubory. Struktura, index-sekvenční přístup ke komponentám, test a nastavení pozice. Módy otevření. Manipulace s bloky dat u netykových souborů. Ošetření chybových stavů vstupně-výstupních operací pomocí funkce *IOResult*.

Cvičení: Předvedení seminárních prací, udělení zápočtů, informace o zkoušce.

TÉMATA PŘEDNÁŠEK A CVIČENÍ V LETNÍM SEMESTRU (II)

15. Typy ukazatel. Typové a netytové (univerzální) ukazatele. Deklarace proměnné a definice hodnoty ukazatele. Segmentová a ofsetová část hodnoty, efektivní adresa, relační operace nad ukazateli. Založení, uvolnění a přístup k dynamické proměnné. Řízení *heapu*. Použití ukazatele k referenci statické proměnné.

Cvičení: Obsluha binárních souborů, demonstrace vnější a vnitřní reprezentace dat. Uživatelské ošetření chybových stavů operací se soubory.

16. Lineární dynamické datové struktury. Jednosměrné spojové seznamy typu zásobník a fronta, uspořádaný seznam. Operace nad spojovými seznamy – inicializace seznamu, vložení, odebrání, zařazení a vyhledání prvku. Použití nárazníku. Obousměrné seznamy.

Cvičení: Použití zásobníku, fronty a uspořádaného seznamu v konkrétních úlohách. Sledování hodnot ukazatelů a dynamických proměnných při krokování programu.

17. Nelineární dynamické datové struktury. Binární stromy. Metody průchodu binárním stromem – „pre-order“, „inorder“ resp. „postorder“. Řazení pomocí stromu, vyvážený strom, vyhledávací strom.

Cvičení: Vytvoření vyhledávacího stromu a jeho použití při vyhledávání. Průběžné vytváření uspořádaného stromu. Postupné uvolnění uzlů stromu.

18. Přetypování proměnné a výrazu. Přístup k proměnné referované netygovým ukazatelem. Netytové parametry podprogramů. Typy procedura. Popis procedurálního typu, význam formálních parametrů v deklaraci. Přiřazení hodnoty procedurální proměnné, volání podprogramu prostřednictvím procedurální proměnné. Aplikace adresních funkcí *Addr*, *Seg* a *Ofs* a adresního operátoru *@* na procedurální proměnné. Procedurální parametry podprogramů.

Cvičení: Ukázka využití přetypování u netypových parametrů podprogramů (test rovnosti proměnných obecných typů dané velikosti). Ukázka využití procedurálních proměnných (pole procedur, volených pomocí indexu) a procedurálních parametrů (vytvoření tabulky hodnot funkce, zadané parametrem). Sledování hodnot procedurálních proměnných, využití přetypování pro sledování hodnot proměnných neznámého typu.

19. Přístup programu k datům v operační paměti. Datový a zásobníkový segment, halda. Globální a lokální statické (deklarované) proměnné a dynamické (zakládáné) proměnné. Řízení velikosti zásobníku a haldy programu. Absolutní proměnné, klauzule *absolute*. Přístup k operační paměti a portům.

Cvičení: Využití absolutní proměnné pro přístup do videopaměti v textovém režimu (animace pohybu znaku). Ukázka využití přístupu k portům (čekání na stisk a uvolnění klávesy Shift).

20. Modulární programování. Účel programových jednotek. Připojení jednotky k programu, klauzule *uses*. Struktura jednotky: hlavička, části interfejsová, implementační a inicializační. Veřejné a privátní deklarace, rozsah jejich platnosti.

Cvičení: Vytvoření programové jednotky (implementace jednoduchých prostředků pro souřadnicově orientovaný výstup textu na displej a podporu barev) a její použití v demonstračním programu.

21. Klauzule *uses* v interfejsové a v implementační části jednotky. Přípustné a nepřípustné kruhové odkazy mezi jednotkami. Hierarchie modulů programu, redeklarace a stínění, kvalifikované identifikátory. Přímá a nepřímá závislost modulů, kompilace závislých modulů. Rozsah a rozložení kódu a dat modulů programu v operační paměti. Záměrná fragmentace kódu rozsáhlého programu využitím inicializačních částí jednotek.

Cvičení: Implementace dalších podprogramů v jednotce z předchozího semináře. Překlad jednotky z okna programu – využití příkazů *Compile*, *Make* a *Build* vývojového prostředí. Ukázka problémů při kompilaci závislých modulů, ukázka pořadí provedení inicializačních částí jednotek vzhledem k pořadí jejich uvedení v klauzuli *uses* programu a vzhledem k jejich závislosti.

22. Jednotka *System*. Prostředky pro ovládání diskových souborů, generátor náhodných čísel, parametry programu, vynucení předčasného ukončení výpočtu. Řízené ukončení běhu programu, implementace ukončovacích procedur. Jednotka *Printer*, ovládací proměnná tiskárny *Lst*. Analogie s proměnnými *Input* a *Output* jednotky *System*.

Cvičení: Práce s parametry programu (např. přejmenování diskového souboru, jehož původní a nová specifikace jsou programu předány z příkazové řádky), zadávání parametrů během ladění a testování programu ve vývojovém prostředí. Ukázka korektní implementace ukončovacích procedur programu a jednotek.

23. Jednotka *Dos*. Ošetření chybových stavů. Ovládání systémových hodin. Zjišťování údajů o souborech a ostatních položkách adresářů, vyhledávání. Spouštění externích programů a volání přerušení.

Cvičení: Využití některých prostředků jednotky *Dos* (vytvoření programu pro výpis prvků zvolené složky – obdoby systémového příkazu *dir*). Ukázka volání externího programu, systémového příkazu a dávkového souboru. Využití výstupního kódu.

24. Jednotka *Crt*. Vstup textu z klávesnice, proměnné *CheckEof* a *CheckBreak*, funkce *ReadKey* a *KeyPressed*. Textové režimy obrazovky, výstupní okna, ovládání barev, souřadnicově orientovaný výstup textu. Struktura videopaměti v textových režimech. Tónový generátor.

Cvičení: Cyklus čekání na stisk libovolné klávesy nebo některé z kláves zvolených, čtení znaku bez odezvy na obrazovce, detekce stisku funkčních kláves a kurzorových s dvojnásobným kódem. Ukázka ovládání tónového generátoru (přehrání posloupnosti tónů podle zápisu jejich parametrů v zadaném souboru).

25. Jednotka *Graph*. Ošetření chyb. Instalace grafického ovladače, přepnutí do grafického režimu. Výstupní okno, souřadná soustava okna, ořezávání, ovládání grafického kurzoru, barvy a palety.

Cvičení: Demonstrace práce s barvami a paletami barev (například vykreslení barevného obrázku a následná manipulace hodnot registrů palety).

26. Jednotka *Graph*. Elementární prvky grafického výstupu: úsečky, oblouky, vyplňované obrazce, grafický text. Styl čar, mód zápisu čar, výplňové vzory, styl textu.

Cvičení: Kreslení obrázků, práce s grafickým textem.

27. Jednotka *Graph*. Operace s výřezy, stránky videopaměti, animace. Připojení ovladačů a písem ke kódu programu. Utilita *binobj.exe*.

Cvičení: Animace pohybu obrázku na barevném pozadí pomocí procedur *GetImage* a *PutImage* v různých režimech zápisu do videopaměti. Demonstrace začlenění grafických ovladačů a písem do kódu programu.

28. Řízení překladu. Přepínačové, parametrické a podmínkové direktivy kompilátoru. Vlivy přepínačových direktiv na bezpečnost, časovou efektivitu a rozsah cílového kódu. Podmíněná kompilace.

Cvičení: Předvedení seminárních prací, udělení zápočtů, informace o zkoušce.

ZÁPOČET A ZKOUŠKA

V obou semestrech musí posluchač získat zápočet a složit zkoušku. Předpokladem pro udělení zápočtu je předvedení seminární práce – samostatně řešeného programu na individuálně zvolené téma, odpovídající úrovni znalostí v daném semestru. Zdrojový text musí být podrobně okomentován, není však třeba jej tisknout.

Zkouška sestává organizačně z písemné a z ústní části. V písemné části posluchač vytvoří program, řešící zadanou úlohu (z oblasti zpracování textových souborů v prvním semestru resp. aplikace dynamických datových struktur v semestru druhém). V ústní části posluchač provede rozbor řešené úlohy a odpoví na doplňující otázky. Požadovaná úroveň znalostí odpovídá rozsahu přednášky.

STUDIJNÍ LITERATURA

POLÁCH, EDUARD: *Programování v jazyku Turbo Pascal*. České Budějovice, PFJU, 1993. ISBN 80-7040-076-5. URL elektronické verze www.pf.jcu.cz/~edpo/program/program.html.

POLÁCH, EDUARD: *Řešené úlohy z Pascalu: Textové soubory*. České Budějovice, PFJU, 1997. Webová stránka. URL www.pf.jcu.cz/~edpo/pas1/pas1.html.

MIKULA, PAVEL – JUHOVÁ, KATEŘINA – SOUKENKA, JIŘÍ: *Borland Pascal 7.0 kompendium*. Praha, Grada, 1994. ISBN 80-7169-009-0.

POUŽÍVANÝ SOFTWARE

V rámci cvičení z programování studenti používají vývojové prostředí *Borland Pascal 7.0*, instalované na školních počítačích. Pro domácí práci lze použít nižší verzi *Turbo Pascal 5.5*, která je volně dostupná na stránkách firmy Borland:

TURBO PASCAL 5.5: <http://community.borland.com/article/0,1410,20803,00.html>

KONTAKT NA UČITELE

RNDr. Eduard Polách. Jeronýmova 10, kacíř J521. E-mail: edpo.pf.jcu.cz. Web: www.pf.jcu.cz/~edpo.