*Реферат з інформатики*

**Програмування: Структури перехід та вибір варіанту, типи символів та перелічувані.**

1

. Мітки та переходи

У мові Паскаль будь-який оператор і кінець складеного оператора (слово end) можна відмітити, тобто ідентифікувати, додати йому індивідуальне ім'я. Це ім'я називається міткою. У авторській версії мови мітками могли бути цілі сталі від 1 до 9999, у мові Борланд-Паскаль до них додано ідентифікатори. Мітка записується перед оператором або словом end через двокрапку, наприклад,

*1 : money := 21;*

 *mmm : money:=0;*

 *finita : end.*

Мітки, використовувані в тілі програми або підпрограми, повинні бути означені в її ж блоці. Означення міток має вигляд:

*label список-міток-через-кому ;*

*наприклад, label 1, mmm, finita*;.

Мітка, означена в блоці, повинна відмічати рівно один оператор у тілі цього блоку.

Мітки використовуються в операторах переходу, що мають вигляд

 *goto мітка*

наприклад,

 *goto 1; if x>1000 then goto mmm.*

Результатом виконання оператора переходу є те, що слідом за ним виконується оператор, відзначений цієюміткою.

У випадку, коли відзначений кінець складеного оператора, можливі варіанти.

Якщо це не кінець тіла циклу і не кінець програми, то буде виконуватися наступний оператор.

Якщо відзначений кінець програми, то її виконання завершується.

Якщо відзначений кінець тіла циклу, то виконуються дії, які слідують за виконанням тіла (перевірка умови продовження while-циклу або неявна зміна параметра for-циклу).

Оператор переходу й відповідний відмічений оператор повинні бути записаними в тілі блоку (програми або підпрограми), де цю мітку означено. Іншими словами,

переходи з одного блоку в інший і використання неозначених у блоці міток заборонено.

Ми не схильні популяризувати використання операторів переходу. Більше того, у свій час було доведено, що без них узагалі можна обійтися (достатньо умовних операторів і while-циклів). Проте є три випадки, коли указання переходу доречно й зручно:

 на кінець підпрограми;

 зсередини циклу на його кінець;

 зсередини циклу на наступний за циклом оператор.

У Турбо Паскаль для указання таких переходів є спеціальні оператори, відповідно*, exit,* *continue і break*, тобто"вийти" (з підпрограми), "продовжувати" і "перервати" (виконання циклу). Розумно використовуючи ці оператори, можна обійтися без міток і goto. Насправді ці три оператори – виклики процедур. Їх імена не є службовими словами виділяти їхнім .

 2. Тип символів

Розглянемо останній з базових типів – тип символів. Множина символів, представних у сучасному комп'ютері, як правило, складається з 256 елементів.

У мові Паскаль символ позначається символьною сталою, що є символом в апострофах: 'A', '1', '.' тощо. Сам символ "апостроф" задається символьною сталою ''''.

Символьна стала – це не символ, а його позначення в мові Паскаль.

Не всі символи позначаються сталими мови Паскаль, але будь-який символ можна задати за допомогою виклику функції chr. Символам у комп'ютері ставляться у відповідність номери від 0 до 255. Якщо i – цілий вираз із значенням від 0 до 255, то вираз вигляду chr(i) задає символ із номером, що є значенням i. Відповідність символів та номерів від 0 до 127 зафіксовано в Американському стандартному коді для обміну інформацією (ASCII). Наприклад, chr(48) позначає те ж саме, що й стала '0', chr(48+1) – то ж саме, що '1', chr(65) є синонімом сталої 'A', chr(97) – сталої 'a'.

Цілий номер символу породжується викликом функції "порядковий номер", тобто виразом вигляду ord(c), де значенням виразу c є символ. Наприклад, ord('0')=48, ord('A')=65, ord('a')=97. За своїм означенням, функції chr і ord взаємно обернені, тобто chr(ord(c))=c за будь-якого символу c, і ord(chr(n))=n за будь-якого n=0, 1, … , 255.

Крім функції ord, для символів означені порівняння, причому

 a<b ? ord(a)<ord(b).

Очевидно, наприклад, що ' ' < '\_' < '0' < '1' < … < '9' < 'A' < 'B' < ... < 'Z' < 'a' < 'b' < … < 'z'.

До символів застосовна операція катенації, або дописування, що позначається знайомим знаком "+". Її результатом є не символ, а послідовність із двох символів, або рядок. Наприклад, '1'+'2' є послідовність символів, яку можна задати літералом '12'.

Множина символів і операції, означені для них, утворюють тип символів, іменований char. Змінні цього типу називаються символьними.

Задачі

1). Указати різницю між 0 і '0', A і 'A', - і '-'.

2)Обчислити значення виразу:

а) chr(ord('0')+9); б) chr(ord('A')+1); в) chr(ord('A')+25);

г) chr(ord('0')-16); д) 'Z'>'a'; е) ord('9')-ord('0').

3)Написати вираз, що задає перевірку, чи є значення символьної змінної ch:

а) цифрою від '0' до '9';;

б) латинською буквою.

4)Написати вираз, що задає обчислення:

а) цілого числа від 0 до 9 за значенням символьної змінної ch від '0' до '9';

б) символу від '0' до '9' за цілим значенням змінної dg від 0 до 9.

5)Написати програму друкування таблиці символів та їх номерів, починаючи з символу ' ' (пропуск), номер якого 32.

3.. Перелічувані типи

Елементи кожного з типів Boolean, char і integer упорядковані відповідно до операції порівняння <. У цьому порядку їх можна взаємно однозначно зіставити натуральним числам 0, 1, … (перелічити їх). За будь-яким натуральним числом можна указати наступне за ним і попереднє перед ним (крім числа 0). Так само за будь-яким елементом, крім найбільшого, цих типів можна назвати наступний за ним і за будь-яким елементом,

крім найменшого, – попередній перед ним. Наприклад, наступним за false є true, наступним за chr(0) – chr(1), за chr(254) – chr(255).

Елементи типу real теж упорядковані, але поставити їм у відповідність натуральні 0, 1, … неможливо, як і за дійсним числом указати наступне за ним. Отже, всім базовим типам, окрім типу real, притаманно те, що для них означено операції "наступний", "попередній" та "порядковий номер".

У мові Паскаль перелічуваним називається тип, для якого означено операції SUCC (наступний), PRED (попередній) і ORD (порядковий номер елемента), а також порівняння =, <>, <, >, <=, >=.

Типи boolean, char і integer, на відміну від типу real, є перелічуваними. Операції succ, pred, ord записуються у вигляді викликів функцій: вирази pred(1), succ('a') і ord(true) мають значення відповідно 0, 'b' і 1.

Всі чотири базових типи називаються скалярними, тому що значення цих типів не складені з частин, які можна було б окремо ідентифікувати. У наступному підрозділі ми розглянемо спосіб означення власних скалярних типів. Але всі вони будуть перелічуваними.

У мові Бор ланд-Паскаль для всіх перелічуваних типів означено операції, аналогічні succ і pred. Вони застосовуються до змінних перелічуваних типів у вигляді викликів процедур INC і DEC. Виклик inc(z) рівносильний оператору z:=succ(z), dec(z) – оператору z:=pred(z). У виклику може бути другий аргумент – цілочисловий вираз. Наприклад, виклик inc(z, 2) задає збільшення z на дві "одиниці" того типу, до якого належить z. Якщо змінна z типу char і має значення 'A', то в результаті виконання inc(z, 2) її значенням буде 'C'.

Значення виразу може бути й від'ємним – тоді z зменшиться. Аналогічно за виконання dec(z, 3) значення z зменшується у його типі, наприклад, від 3 до 0 або від 'D' до 'A'.

4. Означення власних перелічуваних типів

Мова Паскаль має засоби означення власних типів, необхідних у створенні реальних програм. Носій типу, тобто множина його елементів, задається в означенні імені типу. Загальний вигляд означення такий:

*type ім'я = вираз, що описує елементи множини;*

Службове слово type – це англійське "тип". З виразами, що задають множини елементів, ми будемо знайомитися поступово.

Найпростіший вираз – це ім'я типу, уже означене вище або ім'я стандартного типу. Наприклад, означення

*type int = integer;*

задає нове іменування типу integer. Після цього означення скрізь замість довгого імені integer можна писати коротке іnt.

Вирази, що задають множини елементів, можна записувати як в означенні імені типу, так і на місці імені типу в означеннях змінних.

Мова Паскаль дозволяє програмісту означати свої власні перелічувані типи. Вони називаються типами-переліками та типами-діапазонами. Переліки наочно подають множини таких значень, що природно позначаються словами. Наприклад, сигнали світлофора або дні тижня.

Множину днів тижня, позначену {Пн, Вт, Ср, Чт, Пт, Сб, Нд} у мові Паскаль можна задати, означивши тип-перелік з ім'ям, наприклад, Weekd:

*type Weekd=(Mon, Tue, Wed, Thu, Fri, Sat, Sun).*

Тут ужито скорочення від англійських слів Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday –понеділок, вівторок тощо до неділі.

Аналогічно сигнали світлофора можна задати, определив тип Lights:

*type Lights=(red, yellow, green) {червоний, жовтий, зелений}*

Як бачимо, перелік задається виразом вигляду *( список ідентифікаторів* )

Ідентифікатори списку після цього означення виступають сталими відповідного типу. Сталі не можуть повторюватися в означеннях того самого блока. Наприклад, якщо разом із типом сигналів світлофора означити

ще тип кольорів райдуги, то використовувати імена red, yellow, green не можна.

Порядок сталих у списку задає операції ord, pred і succ, означені для всякого переліку. Наприклад, ord(Mon)=0, ord(Tue)=1 тощо, ord(Sun)=6, succ(Mon)=Tue, pred(Sat)=Fri. У відповідності зі значеннями ord означено й порівняння, наприклад, red<yellow, Mon<Sun.

Вираз із сталими перелічуваних типів можна присвоювати однотипним із ними змінним.

Сталі й змінні перелічуваних типів, на відміну від базових типів, не можна вказувати у викликах процедур виведення та введення.

Будь-який перелічуваний тип, стандартний або тип-перелік, означений вище в програмі, може виступати базовим для типу-діапазону. Він задається виразом вигляду

 *стала1..стала2,*

де стала1 <= стала2. Наприклад, ми можемо означити типи

 *type hundred=0..99; chesslet='A'..'H'; Workd=Mon..Fri*

або змінні var stop : red..yellow; column : chesslet тощо.

Для діапазону означено ті самі операції, що для його базового типу.

5.. Оператор вибору варіантів

Розробимо програму "найпростіший калькулятор" . Будемо використовувати змінну signop із значеннями 1, 2, 3 або 4, що познає знаки операцій відповідно "+", "-", "\*", "/". Її значення, отримане в результаті читання, порівнюємо з 1, 2, 3, 4 та виконуємо відповідні їм дії:

*if signop=1 then first:=first+second else*

*if signop=2 then first:=first-second else*

*if signop=3 then first:=first\*second else*

*{signop=4}*

 *first:=first/second;*

Цей оператор задає вибір потрібного оператора з тих, які записано після слів then, залежно від значення змінної signop. Такий вибір у мові Паскаль можна задати інакше, використавши оператор вибору варіантів, або case-оператор (case – англійське "випадок"). У даній програмі він має вигляд:

*case signop of*

*1: first:=first+second;*

*2: first:=first-second;*

*3: first:=first\*second;*

*4: first:=first/second*

*end;*

Вираз після слова case (тут це ім'я signop) називається селектором варіантів. Його значення послідовно порівнюється з числами, що "відмічають" оператори-варіанти. Як тільки значення селектора співпаде з числом, буде виконано відповідний оператор, і все закінчиться. Якщо ж значення селектора відрізняється від усіх чисел-відміток

варіантів, то жодний із варіантів не виконується.

Селектором може бути довільний вираз будь-якого перелічуваного типу. Варіант може бути відмічений списком із кількох сталих відповідного типу; вони записуються через кому. Наприклад, нехай робочий час у робочі дні 8 годин, а у вихідні – 0. За дії означення типу Weekd ми можемо запам'ятати в змінній wt робочий час:

*var dd : Weekd; wt : integer;*

*…*

*case dd of*

*Mon, Tue, Wed,Thu, Fri : wt:=8;*

*Sat, Sun : wt := 0*

*end.*

Значення селектора порівнюється з усіма значеннями в черговому списку, і варіант виконується, якщо воно дорівнює одному зі значень списку. Множини значень у списках можуть перетинатися – буде виконано той варіант, у списку якого вперше знайдене значення селектора.

У мові Бор ланд-Паскаль у списках, що відзначають варіанти, можна записувати не тільки сталі, але й діапазони.

Наприклад, останній оператор можна переписати так:

*case dd of*

*Mon .. Thu, Fri: wt:=8;*

*Sat, Sun: wt:=0*

*end.*

За останнім варіантом можна після слова else указати необов'язковий альтернативний варіант. Цей варіант виконується, якщо значення селектора немає в жодному зі списків. Наприклад, останній оператор можна переписати в такому вигляді:

*case dd ofMon .. Fri: wt:=8;*

*Sun: wt:=0*

*else wt:=0*

*end.*

Задачі

1)Переписати "найпростіший калькулятор із використанням case-оператора і читанням знаків операцій у вигляді символів.

2)У давньояпонському календарі був прийнятий 60-річний цикл із п'яти 12-річних підциклів, послідовно позначених кольорами – зеленим, червоним, жовтим, білим та чорним. У кожному підциклi послідовні роки мали назви тварин: пацюка,

корови, тигра, зайця, дракона, змії, коня, вівці, мавпи, курки, собаки та свині. 4-й рік нової ери – рік зеленого пацюка – був початком чергового циклу.

Написати програму читання року нової ери та друкування його назви за давньояпонським календарем.