Умарова Алия Жумажановна - учитель информатики

СОПШДО № 17 г. Павлодара

Урок № 1, 2. Тема: «Типы данных в Паскаль»

**Цель урока: сформировать у учащихся целостного представления о типах данных языка Паскаль.**

***План урока***

Организационный момент – 10 минут

Лекция – 25 минут

Закрепление – 8 минут

Домашнее задание – 2 минуты

***Ход урока***

I. Организационный момент

Итак, тема урока “Типы данных”. Запишите тему урока.

**II. Лекция (рассчитанная на два занятия с приложением в виде презентации)**

Функционирование любой программы связано с обработкой данных. Данные, предназначенные для обработки, называются исходными и задаются обычно в начале программы. Программа по ходу выполнения может запрашивать недостающие исходные данные.

В процессе выполнения программы исходные данные преобразуются в результаты.

Каждый элемент данных, используемый в программе, является константой или переменной.

Константами называются элементы данных, значения которых в процессе выполнения программы не изменяются. В языке Turbo Pascal используются константы следующих видов: числовые, логические (булевские), символьные и строковые.

Числовые константы предназначены для представления числовых данных (целых и вещественных). Булевские константы используются для представления данных, имеющих смысл логических высказываний (да - нет, истина – ложь, 1 - 0). Символьные и строковые константы – это отдельные символы и их последовательности.

Переменные, в отличие от констант, могут менять свои значения при выполнении программы. В программировании переменную можно трактовать как одну или несколько ячеек оперативной памяти компьютера, которым присвоено определенное имя (идентификатор). Содержимое этих ячеек может меняться, но имя переменной остается неизменным. Каждое новое значение, записанное в ячейку памяти, “затирает” предыдущее значение, поэтому в любой момент времени переменная имеет только одно, текущее, значение. Обычно переменные используются для хранения исходных данных, результатов программы, а также промежуточных данных, которые образуются по ходу выполнения алгоритма.

**Вопрос**. Как вы думаете, может ли быть верным данное высказывание а:=а+1?

В математике значение переменной в рамках определенной задачи неизменно, поэтому математик сочтет это высказывание неверным. Для программиста это абсолютно правильная конструкция, которая задает вычисление суммы содержимого ячейки а и числовой константы 1 и занесение полученного результата в ту же ячейку а. После выполнения этого действия старое значение переменной а будет потеряно, т.к. одна ячейка памяти не может вместить сразу несколько значений. Это очень важный момент в программировании.

Именование констант и переменных в программировании очень похоже на использование символических выражений в алгебре, однако, для того чтобы компилятор смог их обрабатывать, нужно снабдить его некоторой дополнительной информацией – выполнить описание. В этой информации сообщается о типе каждой именованной величины.

Для описания множества допустимых значений величины и совокупности операций, в которых может участвовать данная величина, используется указание ее типа данных. Тип данных (data type) – множество величин, объединенных определенной совокупностью допустимых операций. Каждый тип имеет свой диапазон значений и специальное зарезервированное слово для описания. Все типы данных можно разделить на две группы: скалярные и структурированные (составные). Скалярные типы, в свою очередь, делятся на стандартные и пользовательские.

Перечень типов данных в языке Турбо Паскаль можно представить в виде следующей схемы:

Рисунок 1



Простые (скалярные) типы данных

К скалярным (scalar - простые) типам данных относят типы данных таких величин, значения которых не содержат составных частей.

Все простые данные имеют два характерных свойства: неделимость и упорядоченность их значений.

Целочисленные типы данныхТип Диапазон Требуемая память (байт)

byte 0..255 1

shortint -128..127 1

integer -32768..32767 2

word 0..65535 2

longint -2147483648..2147483647 4

Целочисленные типы данных представляют собой значения, которые могут использоваться в арифметических выражениях и занимать в памяти от 1 до 4 байт.

Вещественные типы данныхТип Диапазон Мантисса Требуемая память (байт)

real 2.9\*10E – 39..1.7\*10E38 11-12 6

single 1.5\*10E – 45..3.4\*10E38 7-8 4

double 5.0\*10E – 324..1.7\*10E308 15-16 8

extended 1.9\*10E – 4951..1.1\*10E4932 19-20 10

comp -2E+63+1..2E+63-1 10-20 8

Вещественные типы данных представляют собой вещественные значения, которые могут использоваться в арифметических выражениях и занимать в памяти от 4 до 6 байт. Паскаль допускает представление вещественных значений и с плавающей запятой, и с фиксированной точкой.

Вещественные значения могут изображаться в форме с фиксированной точкой, например 7.32, 456.721 или 0.015, а также в форме с плавающей точкой, т.е. парой чисел вида <мантисса>Е<порядок> (7.32Е+00, 4.56721Е+02, 1.5Е-02).

Литерный (символьный) тип

Литерный (символьный) тип char определяется множеством значений кодовой таблицы ПЭВМ.

Каждому символу приписывается целое число в диапазоне от 0 до 255. Для размещения в памяти переменной литерного типа требуется один байт.

Булевский тип

Булевским типом называют тип данных, представляемый двумя значениями true (истина) и false (ложь). Он широко применяется в логических выражениях и выражениях отношения. Для размещения в памяти переменной булевского типа требуется 1 байт.

Пользовательские типы

Кроме стандартных типов данных Паскаль поддерживает скалярные типы, определенные самим пользователем. К ним относятся перечисляемый и интервальный типы.

Данные этих типов занимают в памяти один байт, поэтому скалярные пользовательские типы не могут содержать более 256 элементов. Их применение значительно улучшает наглядность программы, делает более легким поиск ошибок, экономит память.

Интервальный тип (диапазон)

Интервальный тип позволяет задавать две константы, определяющие границы диапазона значений для данной переменной. Компилятор при каждой операции с переменной интервального типа генерирует подпрограммы проверки, определяющие, остается ли значение переменной внутри установленного для нее диапазона.

Обе константы должны принадлежать одному из стандартных типов (тип real недопустим). Значение первой константы должно быть значительно меньше значения второй.

Например: 1..12 (номер месяца может принимать значения от 1 до 12) или ‘а’..’я’ (буквы русского алфавита – от а до я)

Перечисляемый тип

Перечисляемый тип (enumerated type) – тип данных, заданных списком принадлежащих ему значений.

Объявление перечисляемого типа описывает множество идентификаторов, которые являются возможными значениями перечисляемого типа. Идентификаторы в описании типа представляют собой константы. Отдельные значения указываются через запятую, а весь список заключается в круглые скобки.

Перечисляемый тип ограничен больше чем интервальный, он задается перечислением своих значений.

Например, в виде строковых констант: color=(red, blue, green, black). В приведенном примере создается новый (нестандартный) тип данных color. Переменные этого типа могут принимать всего 4 значения: red, blue, green, black. Такая возможность создания новых пользовательских типов данных имеется в языке Turbo Pascal.

**Структурированные типы данных**

Структурированные типы данных определяют упорядоченную совокупность скалярных переменных и характеризуются типом своих компонентов.

Структурированные типы данных в отличие от простых задают множества сложных значений с одним общим именем. Можно сказать, что структурные типы определяют некоторый способ образования новых типов из уже имеющихся.

Существует несколько методов структурирования. По способу организации и типу компонентов в сложных типах данных выделяют следующие разновидности: регулярный тип (массивы); комбинированный тип (записи); файловый тип (файлы); множественный тип (множества); строковый тип (строки); в языке Турбо Паскаль версии 6.0 и старше введен объектный тип (объекты).

В отличие от простых типов данных, данные структурированного типа характеризуются множественностью образующих этот тип элементов, т.е. переменная или константа структурированного типа всегда имеет несколько компонентов. Каждый компонент в свою очередь может принадлежать структурированному типу, т.е. возможна вложенность типов.

Все структурированные типы данных требуют отдельного рассмотрения и будут подробно изучены нами в дальнейшем, а сегодня мы только их определим.

Строки

Строка (string) – это последовательность символов кодовой таблицы персонального компьютера. Количество символов в строке может изменяться от 0 до 255.

Массивы

Простые типы определяют различные множества неразделимых значений. В отличие от них структурированные типы задают множества сложных значений, каждое из которых образует совокупность нескольких значений другого типа. В структурных типах выделяют регулярный тип (массивы - array). Название регулярный тип (или ряды) массивы получили за то, что в них объединены однотипные элементы, упорядоченные (урегулированные) по индексам, определяющим положение каждого элемента в массиве.

Множества

Множество (set) – это структурированный тип данных, представляющий собой набор взаимосвязанных по какому-либо признаку или группе признаков объектов, которые можно рассматривать как единое целое. Каждый объект в множестве называется элементом множества. Все элементы множества должны принадлежать одному из скалярных типов, кроме вещественного.

Записи

Для записи комбинации объектов разных типов в Паскале применяется комбинированный тип данных – запись (record). Например, товар на складе описывается следующими величинами: наименование, количество, цена, наличие сертификата качества и т.д. В этом примере наименование – величина типа string, количество – integer, цена – real, наличие сертификата – boolean.

Запись представляет собой наиболее общий и гибкий структурированный тип данных, так как она может быть образована из неоднотипных компонентов и в ней явным образом выражена связь между элементами данных, характеризующими реальный объект.

Файлы

Большие совокупности данных удобно иметь записанными во внешней памяти в виде последовательности сигналов. В Паскале для этих целей предусмотрены специальные объекты – файлы (file). Файлом называется совокупность данных, записанная во внешней памяти под определенным именем.

Рассмотри задачи на объявление переменных и констант скалярного типа.

Приступая к решению задач на объявление данных скалярного типа, следует помнить, что:

каждая переменная программы должна быть объявлена;

объявление переменных помещают в раздел, который начинается словом var; константы помещают в раздел, который начинается словом const; переменные пользовательских типов (перечисляемые и интервальные) объявляют по особой схеме;

В имени переменной можно использовать буквы латинского алфавита и цифры (первым символом должна быть буква);

после инструкции объявления данных рекомендуется указывать назначение переменной или константы;

инструкция объявления констант выглядит так: ИмяКонстанты = значение константы;

***Пример:***

const

min=1; {минимальное значение}

max=54; {максимальное значение}

инструкция объявления переменных выглядит так: имя ИмяПеременной: тип;

var

k1: integer; {количество тетрадей}

k2: byte; {количество карандашей}

c1: real; {цена одной тетради}

инструкция объявления переменных интервального типа помещается в двух разделах type, var и выглядит так:

type

<имя типа>=(<константа1> ..<константа2>);

var

<идентификатор, …>: <имя типа>;

Пример:

type

days=1..31 ; {дни месяца}

var

rabotday: days; {рабочие дни}

vihodday: days; {выходные дни}

инструкция объявления переменных перечисляемого типа помещается в двух разделах type, var и выглядит так:

type

<имя типа>=(<значении 1, значение 2, значение n>);

var

<идентификатор, …>: <имя типа>;

Пример:

type

days=(monday, muesday, wednesday, thursday, friday, saturday, sunday) ; {дни }

var

day: days; {дни недели}

season: (may, april, juin ); {дни отпуска}

Рассмотрим несколько задач на объявление данных скалярного типа.

Задача 1. Объявите переменные, необходимые для вычисления значения функции у=х2.

Учитель. Прежде чем записать инструкцию объявления переменных для вычисления того или иного выражения или функции, необходимо: записать формулу; определить количество переменных, входящих в эту формулу; определить какой тип может быть у этих переменных; определить наиболее приемлемый. Важно помнить, что программа должна занимать минимально памяти компьютера, а это напрямую зависит от типа данных.

Анализ задачи. Сколько переменных входят в данную формулу у=х2? Какое значение имеет переменная х? Какое значение имеет переменная у? Какого типа могут быть переменные? Если переменная х – целого типа, то какого типа должна быть переменная у? Какие варианты инструкции объявления переменной для вычисления данной функции могут быть? Запишем их.

|  |  |  |
| --- | --- | --- |
| Вариант 1.  | Вариант 2. |  Вариант 3. |
| varх:real; {аргумент}у: real; {функция}  | varх:integer;{аргумент}у:integer;{функция} | varх: integer; {аргумент}у:real;{функция} |

Какой из этих вариантов самый приемлемый и почему?

Задача 2. Объявите данные, необходимы для вычисления площади круга.

Анализ задачи. Запишем формулу для вычисления площади круга: s=r2. Сколько переменных входят в данное выражение? Есть ли еще данные, кроме переменных, входящих в данную формулу? Чем является число - переменной или константой? Какое значение имеет переменная r? Какое значение имеет переменная s? Если переменная r – целого типа, то какого типа должна быть переменная s? Запишем инструкцию объявления данных.

const

pi=3.14 ; {число }

var

r: real; {радиус}

s: real; {площадь}

Задача 3. Объявите данные, необходимы для вычисления стоимости покупки, состоящей из нескольких тетрадей, карандашей и линейки.

Анализ задачи. Запишем формулу для вычисления площади круга: s=k1\*c1+k2\*c2+c3. Сколько переменных входят в данную формулу? Какое значение имеют переменные k1, k2? Какое значение имеют переменные с1, с2, с3? Какое значение имеет переменная s? Запишем инструкцию объявления данных.

var

k1: real; {количество тетрадей}

k2: real; {количество карандашей}

с1: real; {цена одной тетради}

с2: real; {цена одного карандаши}

с3: real; {цена линейки}

s: real; {стоимость покупки}

III. Подведение итогов урока

Учитель. На этом уроке вы узнали о типах данных языка Паскаль. Давайте еще раз посмотрим схему, отображающую типы данных. И обобщим услышанное во время лекции.

Ответьте, пожалуйста, на вопросы:

Какие данные различают в языке Паскаль? (Ответ: переменные и константы).

На какие две основные группы можно разделить эти данные? (Ответ: простые (скалярные) и структурированные).

Посмотрите на схему и назовите основные типы скалярного типа данных. (Ответ: литерные, вещественные, целые, логические, пользовательские).

Как выглядят инструкции объявления констант и переменных? (Ответ: const Имя Константы=Значение; var ИмяПеременной: тип;)

Почему рекомендуется указывать назначение данных, используемых в программе? (Ответ: во-первых, это считается хорошим тоном программирования, во-вторых, облегчает чтение программы и отслеживание ее работы).

Учитель. Большое всем спасибо за работу. Очень рада, что вы внимательно слушали, помогали мне в работе. И теперь главное, хорошо подготовиться дома к следующему занятию.

IV. Домашнее задание

Знать материал лекции.

Приготовить таблицы диапазонов целых и вещественных чисел.

Знать зарезервированные слова для скалярного типа данных.

Задачи (по карточкам).

V. Карточка для домашнего задания

Объявите переменные, необходимые для вычисления площади кольца.

Объявите переменные, необходимые для вычисления стоимости покупки, состоящей из нескольких тетрадей и такого же количества обложки.