ПЯВУ. Лекция 5.

Основы программирования.
Промежуточные итоги.

А.М. Задорожный

Вопросы для повторения

1. Что такое “блок” в языке программирования?
2. Как объявить переменную в C#? Является ли объявление переменной исполняемой командой?
3. Какие типы данных затрагивались в лекциях?
4. Какой оператор позволяет выбрать одну из ветвей алгоритма?
5. Что такое цикл в программировании? Какие операторы цикла рассмотрены в предыдущих лекциях?
6. Какого типа данные могут читаться (вводиться) с консоли?

Содержание

1. Типы данных
   a) Числовые типы данных и операции над ними;
   b) Текстовые типы данных;
   c) Булевские данные;

2. Понятие выражения, оператор ?: ;

3. Понятие переменной, область видимости и операция присваивания;

4. Оператор if;

5. Операторы цикла
   a) Операторы while и do-while;
   b) Оператор for;

Тип данных

- Важное понятие
- **int**, **double**, **bool**, **string**, **char**

Тип данных определяет:

- Множество значений, в котором могут принимать значения величины данного типа

- Набор операций, которые можно выполнять над величинами данного типа

Числовые типы данных

- **Int** – область значений целые числа в диапазоне от -2^{30} до 2^{30} ($\sim 2*10^9$).

(Важна ограниченность, а не конкретные значения)

Операции над целыми:

=, +, -, *, /, %, &, |, ^, ~, *сравнения, …*

- **Double** – вещественные числа с точностью 17 знаков в диапазоне ~ от 10^{200} до 10^{-200}.

=, +, -, *, /, *сравнения, …*

Числовые типы данных. Свойства.

- **Int**
  - Ограниченный диапазон;
  - Точные вычисления;
  - Побитовые операции.

Литералы: 1, 99, …

Предназначены для управления работой программы (количества итераций), выбора элементов (символ в строке), …

Числовые типы данных. Свойства.

- **Double**
  - Огромный диапазон
  - Высокая, но ограниченная точность

Литералы: 1.0, 2.57 …

Область применения – расчеты

Помимо встроенных операций много операций из математической библиотеки **Math**.

Числовые типы данных. Преобразования.

- **Int** в **double** может преобразовываться неявно (безо всякого указания)
- В смешанных выражениях всегда преобразуется к высшему типу (**double**)
- **Double** в **int** можно преобразовать только явным образом. При этом, дробная часть будет отброшена. **int** n **= (int) x;**

Операция преобразования типа - (<тип>)

```
int x = 3;
double y = x;        // double y = (double)x;
x = (int)(y/2);      // x == 1

x = (int)(y/2 + 0.5); // округление! Можно Math.Round()
```

Операции в духе C

x += y; // x = x + y

И все другие: -=, *=, /=, %=, &=, |=

Для целых типов операции **++** и **--**;

```
string s = “12345”;
for(int i = 0; i < s.Length; i++)
{
…
}
```

Операции в духе C

Операции присваивания являются выражениями

```
x = y = z = 1; // x = (y = (z = 1));
```

Операция ?:

```
x = x < 0 ? –x : x;   // x = Math.Abs(x);
```

В отличие от

```
if( x < 0)
    x = -1;
```

```
x < 0? –x : x
```

является **выражением**

Выражение

Выражение – языковая конструкция, которая может быть вычислена и, в результате вычисления, принимает значение определенного типа.

Примеры:

Оператор if не валяется выражением.

Оператор = является выражением.

Контрольные вопросы

1. О чем следует рассказать на вопрос в билете: “**Тип данных abc.**”?

2. Каковы типичные применения типа **int**? Почему?
3. Каковы типичные применения типа **double**? Почему?
4. Почему ограничена точность **double**?

5. Что означает (**int**) в x = (**int**) (y+0.9)? Чему равен x в зависимости от значения y?

6. Какого типа значения будут сравниваться в последней операции?
   int x = 1;
   double y = 1;
   bool f = x == y;

Контрольные вопросы

1. Какое значение примет переменная b в следующем коде?* ☺️

```
int x = -2 000 000 000, y = 2 000 000 000;
bool b = x>y;
```

1. Сколько преобразований типа в следующем коде?

```
double y = 3, x;
x = (int)(y/2 + 0.5);
```

Чему равно x?

2. Сколько преобразований типа в следующем коде?

```
int x = 3;
double y = (x/2);
```

Чему равно y?

3. Что такое “выражение”? Что означает, что оператор ‘?:’ (или любой другой) является выражением?

Текстовые типы данных

char - любой символ.
Операции над символами:
=, ==, !=.
Литералы: ‘ ’, ‘\n’…

String - *все* последовательности символов.
=, +, +=, ==, !=, .Length, [<номер>]
Литералы: “”, “\t”…

Строка – неизменяемый тип.
Если нужно изменить строку, то строится новая строка!

Связь текстовых и других типов данных

.ToString() – возвращает текстовое представление объекта (переменной).

У string имеется метод Format.

String s = **string**.Format(“{0}”, <объект>);

V

Console.WrilteLine(“{0}”, <объект>);

Связь текстовых и других типов данных

У ряда “встроенных” типов имеется метод Parse.

```
int x = int.Parse(s); // Ошибка, если нельзя
double y = double.Parse(s); // Ошибка, если нельзя
```

```
int x;
bool res = int.TryParse(s, out x); // true, false

double y;
res = double.TryParse(s, out y); // true, false
```

Булевские данные

bool – ‘Истина’ и ‘Ложь’

=, &&, ||, ^, !, ==, !=

Литералы: **true**, **false**

Таблица истинности как средство выяснения тождественности

Переменные

Переменная – именованная область памяти, которая служит для временного хранения данных.

память компьютера

Переменная характеризуется:

1. Именем;
2. Типом;
3. Значением;
4. Имеет адрес (и место) в памяти.

Объявление и использование переменных

```
int x = 3;
```

<тип> <имя> = <начальное значение>;

Переменную можно использовать в тексте программы (получать или изменять ее значение) после того как она объявлена.

Задавать при объявлении начальное значение не обязательно. Его можно задать позже оператором присваивания.

Но задать значение нужно обязательно до использования переменной!

```
int x ;
…
x = 3;
```

Блок и область видимости переменных

БЛОК – кода в C# - это часть инструкций, ограниченная {}.

Блок воспринимается как ОДИН оператор. Блок не является *выражением*.

Блоки внутри другого блока называются “вложенными”.

Область видимости переменной ограничена блоком, в котором она объявлена (и вложенными блоками).

Пример объявления и использования переменной

```
int x = 1;
if( x > 0)
{
    int y = x;
}
else
{
    int y = -x;
}
// y объявлена в 2-х блоках. Т.е. имеются 2 разных y!

// Здесь y использовать нельзя!
// а x можно!
```

Операция присваивания

= - служит для изменения значения переменных

<имя переменной> = <выражение совместимого типа>;

Задача обмена значениями двух переменных одного типа.

Контрольные вопросы

1. Какие операции сравнения можно выполнять над строками?
2. Опишите, как происходит сравнение строк?
3. Опишите, что означает сложение строк?
4. Приведите 2 примера выражений, когда тип результата не совпадает с типами ни одного из аргументов выражения.

5. Назовите 4 обязательных свойства переменной
6. Имеют ли значение имена переменных для выполняемой программы (.exe)?
7. Что означает термин приведение (преобразование) типа данных?

8. В чем различие между числом и цифрой? Как преобразовать число в цифры?
9. Как преобразовать текстовое представление числа в число?
10. Что делает данный код? Зачем sum объявлена перед оператором for?

```
int sum = 0;
for(int i=1; i<N; i = i + 1)
{
        sum = sum + i;
}
```

Операторы

```
if(<условие>)
{
    …
}
else
{
    …
}
```

Операторы

Оператор с предусловием

```
while (<условие>)
{
    <тело цикла>
}
```

Оператор с постусловием

```
do
{
    <тело цикла>
} while (<условие>);
```

Операторы

```
for(<инициализатор>;<условие>;<итератор
>)
{
    тело цикла
}
```

Пример

Задача. Заменить в исходной строке все пробелы на знак подчеркивания.

```
string s = Console.ReadLine();
string t = “”;
for( int i = 0; i < s.Length; i++)
    t += s[i] == ‘ ‘? ‘_’: s[i];
```

Какая часть примера составляет РЕШЕНИЕ задачи, а какая подготовку к решению?

Контрольные вопросы

1. Сколько операторов мы изучили? Какие?
2. К какому типу операторов цикла относятся операторы while, do while и for?
3. Из скольки (и каких) частей состоит управляющая строка оператора for?
4. Если в инициализаторе оператора for объявлена переменная, то какова область ее видимости?
5. Оператор do-while заканчивается ‘;’ - while (<условие>); Как будет выполняться такая программа (обратите внимание на ; в первой строке):

```
while (<условие>);
{
      <тело цикла>
}
```

Заключение. Часть I.

1. Тип данных. Свойства *bool*, *int*, *double*, *string* и *char*. Литералы. Преобразования типов.

2. Переменные: объявление, область видимости и использование..

3. Операции: =, + … , == …, += …, ++/--, && …, *побитовые*| …, [].

4. Инструкции и операторы: *';'*, *if else*, *while*, *do while*, *for*, блок {} и *?:*.

5. Операции ввода-выводя для консоли.

6. Математическая библиотека Math.

Заключение. Часть II.

1. Структура простейшей программы на C#. Текст программы, компиляция и выполняемая программа.
2. Модель компьютера (процессор, память, шина).
3. Двоичное представление **целых** чисел.
4. Представление чисел с плавающей точкой.
5. Некоторые приемы программирования и простейшие алгоритмы в примерах.
6. Терминология.