Программирование на C#

Семинар №2

Модуль № 1

Тема:

Типы данных. Переменные и операторы

Задачи с переменными простых арифметических типов.

Ввод и преобразование типа.

Parse() и TryParse(), простой обработчик исключений и проверка корректности ввода данных.

Арифметические операции.

Задания преподавателя к семинару:

1. Ответьте на вопросы со знаком ToDo.
2. Изучите листинг программы ReversNumber (перевернутое число)
3. Выполните задания со знаком Self.

Задания повышенной сложности со знаком *

Полезные материалы к семинару

1.Таблица типов с плавающей запятой.
https://learn.microsoft.com/ru-ru/previous-versions/visualstudio/visual-studio-2008/9ahet949(v=vs.90)?redirectedfrom=MSDN

2.Таблица целых типов

https://learn.microsoft.com/ru-ru/previous-versions/visualstudio/visual-studio-2008/exx3b86w(v=vs.90)

3.Операторы языка C#

https://msdn.microsoft.com/ru-ru/library/6a71f45d.aspx

Типы данных

C# - язык со строгой статической типизаций (тип переменной определяется при объявлении):

Инициализация строковым литералом

```
int a, b=2;        string str;        string strint = “Hello!”;
        double c, k=2.22;   char ch = ‘#’;
```

Инициализация символьной константой

Внимание! При вводе вещественного числа с клавиатуры целая часть отделяется от дробной запятой, в коде – точкой.

Анонимный тип (при объявлении переменной с типом var) тип выводится по типу выражения справа в операции присваивания:

```
var prim1;
prim1 = 222; // prim1 – тип int
```

Выясним тип объекта:

string st = prim1.GetTipe();

Целочисленные литералы

Десятичные
±XXX, где X - десятичная цифра
Шестнадцатеричные
±**0x**YYY,где Y – 16-ричная цифра
Двоичные / бинарные
±**0b**YYY,где Y – 2-чная цифра

Символ подчёркивания допустим как разделитель цифр:
0xFF_FF_FF_FF
0b_0010_1010

Возможно указание типа литерала явно:
250L – имеет тип long (L-это суффикс)
250U – беззнаковая (U – суффикс)
250UL – беззнаковая длинная (L - суффикс)

Примеры объявления переменных с инициализацией:

```
int a, b=10, c=int.Parse(“355”), d=b+c, e=short.MaxValue;
```

Максимальное значение диапазона для типа short

Вещественные литералы

форма F ±XXX.YYY

форма E ±МантиссаE ±Порядок

Примеры литералов:

5.5 => 5,5 5.0 => 5,0 5. => 5,0

0.5 => 0,5 .5 => 0,5 2.5 E2 => 2,5*102 => 250,0

Все литералы имеют тип данных double. Тип литерала, в котором используется суффикс F - float:

25.5 => double

25.5F => float

Примеры :

double a, d=.5, c=double.Parse(“5.25”);

Ошибка! Следует использовать запятую!

Но это зависит от параметров локализации всей системы!

Литералы

На один символ отводится область памяти длиной 2 байта. (кодировка Unicode).

Пример записи литералов:

Пример: ‘A’ или ´\u0041’ - это код латинской буквы А

Можно с помощью методов (возвращают true/false) выяснить принадлежность кода к группе символов:

IsDigit() десятичная цифра

IsLetter() буква

IsLetterOrDigit() цифра или буква

IsLower() строчная буква

IsUpper() прописная буква

Конвертация: **ToLower(символ)** – прописные -> в строчные

ToUpper (символ) - строчные -> в прописные

Операнды и выражения

Операнды:

- Вызов метода
- Переменная
- Константа
- Выражение

Выражения – фраза на языке программирования, предназначенная для выполнения вычислений, которые позволяют получить некоторый результат (значение выражения)

Определение фразы было на лекции

Demo 1. Задачи с переменными простых арифметических типов.

Ввести с клавиатуры натуральное трехзначное десятичное число. Вычислить и вывести на экран число, полученное путем переворота цифр исходного числа. Например: для 253 получить 352
Внимание!!! Алгоритм основывается на известном заранее количестве цифр в исходном числе и ориентирован на накопление всех цифр числа.

Demo 1. Реализация

```
1.    uint  ch,      // Исходное число
2.          copych,   // Копия исходного числа
3.          c1,    c2,    c3,      //Переменные для запоминания цифр
4.                        newch;    // Новое число (после переворота цифр)
5.         string str;    // Строка для приема данных и вывода данных
6.           Console.Write("Введите трехзначное натуральное число:  ");
7.           str = Console.ReadLine();
8.           ch = uint.Parse(str);
9.           copych = ch;
10.          c1 = copych % 10;
11.          copych /= 10;
12.          c2 = copych % 10;
13.          copych /= 10;
14.          c3 = copych % 10;
15.          newch = c1*100 + c2*10 + c3;
16.        str = "Если перевернуть " + ch.ToString() + " будет " + newch.ToString();
17.  //Формирование выходной строки можно реализовать более компактно.
18.           //str = "Если перевернуть " + ch + " будет " + newch;
19.          Console.WriteLine(str);
20.          Console.ReadLine();
```

Особенности целочисленного деления, поэкспериментируем совместно

```
int a, b;
double c;
a=10/4;      //a=2
c=10/4.;     //c=2.5
b=10./4;     //ошибка!!! (10./4.=2.5)
```

Каков знак операции «остаток от деления»?

```
int a;
a=10%3;
a=-10%3;
a=10%-3;
a=-10%-3;
```

Self Выполнить самостоятельно

Self 01 Создайте проект. Объявите переменные типов int, double и char с инициализатором литералом. Выведите на экран каждую переменную с новой строки.

Определите, действительно ли переменные принадлежат к соответствующим типам. Выведите на экран результат.

Self 02 Создайте проект и реализуйте задачу: вычислить сумму целых частей двух вещественных чисел, сумму дробных частей. Вывести на экран полученные значения с комментарием вида:

«Сумма целых частей чисел 25.5 и 12.2 равна 37»

«Сумма дробных частей чисел 25.5 и 12.2 равна 0.7»

Операции инкремента и декремента

Префиксные операции: ++a; --b;

Постфиксные операции: a++; b--;

Определите результат вычисления выражения:

```
int x = 2;

int c = ++x * x++;
```

Demo 2. Принцип Работы Тернарной операции


```
int helper1 = x < y ? (z < x ? z : x) : (y < z ? y : z);
```

Todo 01: Самостоятельно определите значение helper2 и обновите y.

```
int helper3 = x > y ? (z > x ? z : x) : (y > z ? y : z);
x = helper1;
z = helper3;
```

Допишите программу до работоспособного состояния

Self 03. Тернарная Условная Операция

Без использования оператора if:

Напишите программу, меняющую местами значения трех целочисленных переменных x, y, z, так, чтобы в результате его вызова выполнялось требование: **x <= y <= z**.

Введите значения x, y, z, затем упорядочите их и выведите результат в отформатированном виде.

Self 04: Определение Символа

Без использования оператора if:

Напишите программу, определяющую, является ли введённый пользователем с клавиатуры символ цифрой, буквой русского алфавита (прописной либо строчной), или ни тем и ни другим.

В основной программе вводятся символы (тип char), после чего для них с помощью метода выводится соответствующее сообщение:

1. «Это цифра»;
2. «Это буква»;
3. «Это ни буква, ни цифра».

Для проверки условий используйте тернарную операцию. Попробуйте реализовать проверку кодов самостоятельно, без библиотечных методов.

Self 05. Точка Внутри Круга

Без использования оператора if:

Задан круг с центром в начале координат и радиусом R типа double, вводимым с клавиатуры.

Введите координаты точки x и y, и выведите сообщение формата:

«Точка внутри круга!» или «Точка вне круга!»

в зависимости от результата.

Организуйте проверку входных данных и цикл с повтором решения аналогично предыдущим задачам.

Demo 3 . Битовые операции

Установка самого правого бита

```
sbyte a = 127;
        Console.WriteLine(a | (sbyte)(a + 1));
```

Эксперимент: что будет, если убрать приведение типа?

Быстрое умножение и деление числа a на степень двойки n

```
a << n
a >> n
```

Self 06. Вывод Цифр Столбиком

Напишите программу, которая считывает с клавиатуры трёхзначное число типа int и печатает его цифры в порядке от старшего разряда к младшему в столбик.

Пример – для числа 379 вывод будет иметь вид:

3
7
9

Обратите внимание, что для отрицательных трёхзначных чисел цифры **НЕ** должны выводиться со знаком минус.