



ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ
ГОРОДА МОСКВЫ
Государственное бюджетное профессиональное
образовательное учреждение города Москвы
«Колледж малого бизнеса № 4»
(ГБПОУ КМБ № 4)

Отчёт по домашней работе

Специальность: 09.02.07 Информационные системы и программирование

Форма обучения: очная

Студент: Межибор Ярослав Евгеньевич

Группа: ИПО-21.24

Проверил: Рыбаков Александр Сергеевич

Москва, 2025 г.

Переменные

Задание 1

Ответ: Программа завершит выполнение с ошибкой, так как C# является регистрозависимым языком, и переменная была объявлена как name (с маленькой буквы), а в Console.WriteLine используется Name (с большой буквы).

Задание 2

Ответ: Выведет Sam. Сначала переменной person присваивается значение "Tom", а затем значение переменной перезаписывается на "Sam". Вывод на консоль покажет последнее присвоенное значение.

Задание 3

Ответ: Корректными определениями переменных являются:

- string person = "Tom"; (Определение и инициализация)
- string person; (Только определение)

Задание 4

Ответ: Тип, имя, значение.

Задание 5

Ответ: Определение устанавливает тип и имя переменной, а инициализация задаёт начальное значение.

Задание 6

Ответ: C# регистрозависимый язык, поэтому name и Name — разные переменные.

Задание 7

Ответ: Константа инициализируется при определении и её значение нельзя изменить, в отличие от переменной.

Литералы

Задание 1

Ответ: Логические, целочисленные, вещественные, символьные, строковые и null.

Задание 2

Ответ: Вещественные числа с фиксированной запятой и в экспоненциальной форме MЕр

Базовые типы данных

Задание 1

Ответ: Не являются встроенными типами: real, int128, float64

Задание 2

Ответ: bool (логический тип).

Задание 3

Ответ: float (тип с плавающей точкой одинарной точности).

Задание 4

Ответ: uint занимает 4 байта (32 бита).

Задание 5

- string person = "Tom"; (Явное указание типа с инициализацией)
- var person = "Tom"; (Неявное указание типа с инициализацией)
- string person; (Явное указание типа без инициализации)

Задание 6

Ответ: System.Int32, 4 байта.

Задание 7

Ответ: F/f — для float, M/m — для decimal.

Задание 8

Ответ: При использовании var тип переменной определяется автоматически на основе присвоенного значения.

Консольный ввод-вывод

Задание 1

Ответ: Правильный синтаксис с интерполяцией:
Console.WriteLine(\$"Имя: {name} Возраст: {age} Рост: {height} м").

Задание 2

Ответ: Плейсхолдеры — это числа в фигурных скобках (например, \${0}\$, \${1}\$), которые заменяются значениями при выводе на консоль (например, при использовании Console.WriteLine("Имя: {0}", name)).

Задание 3

Ответ: Console.Write() не добавляет переход на следующую строку после вывода, а Console.WriteLine() добавляет.

Задание 4

Ответ: Методом Console.ReadLine(), возвращается в виде строки (string).

Задание 5

Ответ: Методы из класса Convert (например, Convert.ToInt32(), Convert.ToDouble(), Convert.ToDecimal()) и методы Parse() (например, int.Parse(), double.Parse()).

Операции

Задание 1

Ответ: Приоритет операций (от высшего к низшему, для используемых операций):

- Пост-декремент (--)
- Умножение (*), Деление (/), Остаток от деления (%) - выполняются слева направо.
- Сложение (+), Вычитание (-) - выполняются слева направо.
- Присваивание (=)

Шаги выполнения:

Исходное состояние: n1 = 2, n2 = 5.

result = 5 * 3 + 20 / 2 * 2

1.

Пост-декремент (`n1--`): Используется значение `n1=2`, затем `n1` уменьшается до 1.

`result = 5 * 3 + 20 / 2 * 2`

(Теперь `n1 = 1`)

- Умножение (`5 * 3`):

`result = 15 + 20 / 2 * 2`

- Деление (`20 / 2`):

`result = 15 + 10 * 2`

- Умножение (`10 * 2`):

`result = 15 + 20`

- Сложение (`15 + 20`):

`result = 35`

- Присваивание (`result = 35`):

`result = 35`

- Конечный результат: `result = 35`, `n1 = 1`.

Задание 2

Ответ: Приоритет операций (от высшего к низшему, для используемых операций):

- Умножение (*), Деление (/), Остаток от деления (%) - выполняются слева направо.
- Сложение (+) - выполняется слева направо.
- Составное присваивание (+=).

Исходное состояние: `num1 = 4`, `num2 = 5`, `num3 = 15`, `num4 = 10`, `num5 = 5`, `result = 12`.

Выражение: `result = result + (num1 * num2 + num3 % num4 / num5)`

1.

- Умножение (`num1 * num2`): $4 * 5 = 20$

`result += 20 + 15 % 10 / 5$$`

- Остаток от деления ($15 \% 10$): $15 / 10 = 5$

`result += 20 + 5 / 5`

- Деление ($5 / 5$): $5 / 5 = 1$

`result += 20 + 1`

- Сложение ($20 + 1$):

`result += 21`

- Составное присваивание (`result += 21`): $12 + 21 = 33$

`result = 33`

- Конечный результат: `result = 33.`

Задание 3

Ответ:

- Пре-инкремент (`++y`): `y` увеличивается до 10, и это новое значение используется в выражении.

`int z = x++ + 10`

(Теперь `y = 10`)

- Пост-инкремент (`x++`): Используется текущее значение `x=8` в выражении, затем `x` увеличивается до 9.

`int z = 8 + 10`

(Теперь `x = 9`)

- Сложение ($8 + 10$):

`int z = 18`

- Конечный результат: `z = 18.`

`x` будет равна 9, `y` будет равна 10.

Практическое задание

Задание 1

```
1 using System;
2
3 public class SimpleCalculatorNoChecks
4 {
5     public static void Main(string[] args)
6     {
7
8         // Ввод данных
9         Console.WriteLine("Введите первое число (A): ");
10        // Полное преобразование без обработки FormatException
11        double num1 = Convert.ToDouble(Console.ReadLine());
12
13        Console.WriteLine("Введите второе число (B): ");
14        double num2 = Convert.ToDouble(Console.ReadLine());
15
16        // Выполнение и вывод результатов операций
17
18        // Сложение (+)
19        Console.WriteLine($"***Сложение (+):: {num1} + {num2} = {num1 + num2}");
20
21        // Вычитание (-)
22        Console.WriteLine($"***Вычитание (-):: {num1} - {num2} = {num1 - num2}");
23
24        // Умножение (*)
25        Console.WriteLine($"***Умножение (*):: {num1} * {num2} = {num1 * num2}");
26
27        // Деление (/)
28        Console.WriteLine($"***Деление (/):: {num1} / {num2} = {num1 / num2}");
29
30        // Остаток от деления (%)
31        int int_num1 = (int)num1;
32        int int_num2 = (int)num2;
33        Console.WriteLine($"***Остаток от деления (%):: {int_num1} % {int_num2} = {int_num1 % int_num2}");
34
35
36        Console.WriteLine("In--- Результаты Икремента/Декремента ---");
37
38        // Выполнение и вывод Икремента/Декремента
39
40        // Создаем конец для демонстрации эффекта
41        double inc_num1 = num1;
42        double dec_num2 = num2;
43
44        // Икремент (++)
45        inc_num1++;
46        Console.WriteLine($"***Икремент (++):: A + 1 ({num1} + 1) = {inc_num1}");
47
48        // Декремент (--)
49        dec_num2--;
50        Console.WriteLine($"***Декремент (--):: B - 1 ({num2} - 1) = {dec_num2}");
51    }
52 }
```

--- ??????? ?????????? (C#) ??? ??????? ?????? ---
?????? ?????? ?????? (A): 6
?????? ?????? ?????? (B): 8
--- ?????????? ?????? ---
**????????? (+):: 6 + 8 = 14
**????????? (+):: 6 + 8 = -2
**????????? (*):: 6 * 8 = 48
**????????? (*):: 6 * 8 = 0.75
**????????? ?? ?????? (%):: 6 % 8 = 6
--- ?????????? ??????????/????????? ---
**????????? (+):: A + 1 (6 + 1) = 7
**????????? (-):: B - 1 (8 - 1) = 7
--- Code Execution Successful ---