



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

## **Домашняя работа**

**Специальность: 09.02.07 Основы алгоритмизации и  
программирования**

**Форма обучения: очная**

**Студент: Носов Антон Борисович**

**Группа: ИПО 21.24**

**Руководитель: Александр Сергеевич Рыбаков**

**Москва, 2025 г.**

## Тема: “Переменные”

### Вопрос 1:

Что выведет следующий код:

```
string name = "Tom";  
Console.WriteLine(Name);
```

**Ответ: ошибка компиляции**

### Вопрос 2:

Что выведет на консоль следующий код:

```
string person = "Tom";  
person = "Sam";  
Console.WriteLine(person);
```

Варианты ответов:

-Tom

-Sam

-person

**Ответ: Sam**

### Вопрос 3

Какие из следующих вариантов представляют корректное определение переменных:

-string person = "Tom";

-person = "Tom";

-string person;

-string "Tom";

**Ответ:**

**string person = "Tom";**

**person = "Tom";**

**Вопрос 4:**

Какие три основных компонента имеет переменная в языке C#?:

-класс, имя, метод

-тип, размер, область видимости

-имя, индекс, значение

-Тип, имя, значение

**Ответ: 4. Тип, имя, значение**

**Вопрос 5:**

В чём заключается различие между определением переменной и её инициализацией в C#?

-определение создаёт новую переменную в памяти, а инициализация её удаляет.

-определение задаёт начальное значение, а инициализация устанавливает тип переменной.

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

-определение и инициализация — это одно и то же действие.

**Ответ: 3. Определение устанавливает тип и имя переменной, а инициализация задаёт начальное значение.**

#### **Вопрос 6:**

-Почему важно учитывать регистрозависимость при работе с переменными в C#? Приведите пример.

-Регистр важен для типов данных, а не для имён переменных.

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

-В C# регистр не имеет значения для имён переменных.

-Имена переменных в C# должны быть записаны только строчными буквами.

**Ответ: 2. C# регистрозависимый язык, поэтому name и Name – разные переменные.**

#### **Вопрос 7:**

В чём состоит ключевое отличие константы от переменной в C# и как это отражается на их использовании в программе?

-Значение переменной фиксируется при определении и не может быть изменено.

-Константа может быть изменена в процессе работы программы, как и переменная.

-Переменные и константы в C# ничем не отличаются друг от друга.

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

**Ответ: 4. Константа инициализируется при определении и её значение нельзя изменить, в отличие от переменной.**

## Литералы

### Вопрос 1:

Какие виды литералов существуют и чем они отличаются друг от друга?

-Логические, целочисленные, вещественные, символьные, строковые и null.

-целые, дробные, текстовые, булевы и специальные.

-положительные, отрицательные, дробные, символьные и строковые.

-числовые, буквенные, логические, графические и пустые.

**Ответ: 1. Логические, целочисленные, вещественные, символьные, строковые и null.**

### Вопрос 2:

В каких формах могут быть представлены вещественные литералы и как они интерпретируются?

-строковые литералы в двойных кавычках

-Вещественные числа с фиксированной запятой и в экспоненциальной форме MEp

-целые числа в десятичной, шестнадцатеричной и двоичной форме

-символьные литералы в одинарных кавычках

**Ответ: 2. Вещественные числа с фиксированной запятой и в экспоненциальной форме МЕр.**

## **Базовые типы данных**

**Вопрос 1:**

Какие из нижеперечисленных НЕ являются встроенными типами языка C#?

-uint

-sbyte

-real

-int128

-object

-float64

**Ответ: real, int128, float64**

**Вопрос 2:**

Какой тип данных языка C# будет представлять следующая переменная?

bool enabled = true;

**Ответ: Логический тип данных**

**Вопрос 3:**

Какой тип данных языка C# будет представлять следующая переменная?

```
var weight = 84.45f;
```

**Ответ: Неявный тип переменной**

**Вопрос 4:**

Сколько байт занимает значение типа uint?

**Ответ: 4 байта**

**Вопрос 5:**

Какие из следующих вариантов представляют корректное определение переменных:

-string person = "Tom";

-var person = "Tom";

-var person;

-string person;

**Правильные варианты: 1, 2, 3**

**Вопрос 6:**

Какой системный тип соответствует базовому типу данных int в языке C# и сколько байт он занимает?

-System.Int32, 4 байта

-System.Single, 4 байта

-System.UInt32, 8 байт

-System.Int16, 2 байта

**Ответ: 1. System.Int32, 4 байта**

### **Вопрос 7:**

Какие суффиксы используются в C# для явного указания типа данных float и decimal при присвоении значений?

-S/s — для float, D/d — для decimal

-X/x — для float, Y/y — для decimal

-F/f — для float, M/m — для decimal

-L/l — для float, U/u — для decimal

**Ответ: 3. F/f — для float, M/m — для decimal**

### **Вопрос 8:**

Чем отличается объявление переменной с использованием var от явного указания типа данных, например, int?

-var и int — это синонимы для объявления целочисленных переменных.

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

-var используется для объявления переменных с типом string.

-var позволяет объявлять переменные без указания типа и инициализации.



**Ответ: 2. При использовании var тип переменной определяется автоматически на основе присвоенного значения.**

## **Консольный ввод-вывод**

### **Вопрос 1:**

Как вывести на консоль значения нескольких переменных в одной строке с помощью интерполяции?

-Console.WriteLine("{name} {age} {height}");

-Console.WriteLine("Имя: " name " Возраст: " age " Рост: " height "м");

-Console.WriteLine("Имя: {name} Возраст: {age} Рост: {height}м");

-Console.Write(name, age, height);

**Ответ: 3. Console.WriteLine("Имя: {name} Возраст: {age} Рост: {height}м");**

### **Вопрос 2:**

Что такое плейсхолдеры в контексте вывода данных на консоль и как они используются?

-Плейсхолдеры — это числа в фигурных скобках, которые заменяются значениями при выводе на консоль.

-плейсхолдеры используются для создания пустых строк в выводе.

-плейсхолдеры — это имена переменных, которые выводятся на консоль без изменений.

-плейсхолдеры — это специальные символы для форматирования строк.

**Ответ: 1. Плейсхолдеры — это числа в фигурных скобках, которые заменяются значениями при выводе на консоль.**

### **Вопрос 3:**

В чём отличие метода `Console.Write()` от `Console.WriteLine()`?

-`Console.Write()` используется для ввода данных, а `Console.WriteLine()` — для вывода.

-`Console.Write()` выводит информацию в виде таблицы, а `Console.WriteLine()` — в виде списка.

-`Console.Write()` не добавляет переход на следующую строку, а `Console.WriteLine()` добавляет.

-`Console.Write()` может выводить только числа, а `Console.WriteLine()` — любые данные.

**Ответ: 3. `Console.Write()` не добавляет переход на следующую строку, а `Console.WriteLine()` добавляет.**

### **Вопрос 4:**

Каким методом можно получить ввод с консоли и в каком виде он возвращается?

-методом `Console.WriteLine()`, возвращается в виде числа.

-методом `Console.Write()`, возвращается в виде массива.

-методом `Convert.ToInt()`, возвращается в виде строки.

-методом `Console.ReadLine()`, возвращается в виде строки.

**Ответ: 4. Методом `Console.ReadLine()`, возвращается в виде строки.**

### Вопрос 5:

Какие методы предоставляет платформа .NET для преобразования строковых значений в числовые типы данных?

-Convert.ToString(), Convert.ToInt(), Convert.ToChar()

-Parse.ToInt(), Parse.ToFloat(), Parse.ToNumber()

-Convert.ToInt(), Convert.ToDouble(), Convert.ToDecimal()

-Console.WriteLine(), Console.Write(), Console.ReadLine()

**Ответ: 3. Convert.ToInt(), Convert.ToDouble(), Convert.ToDecimal()**

## Операции

### Вопрос 1:

Есть следующий код:

```
int n1 = 2;
```

```
int n2 = 5;
```

```
int result = n2 * 3 + 20 / 2 * n1--;
```

Используя приоритеты операций, разложите выражение `int result = n2 * 3 + 20 / 2 * n1--` по шагам.

**Ответ:**

**5 \* 3 = 15**

**20 / 2 = 10**

**10 \* 2 = 20**

**15 + 20 = 35**

### Вопрос 2:

Есть следующий код:

```
int num1 = 4;  
int num2 = 5;  
int num3 = 15;  
int num4 = 10;  
int num5 = 5;  
int result = 12;
```

```
result += num1 * num2 + num3 % num4 / num5;
```

Используя приоритеты операций, разложите выражение `result += num1 * num2 + num3 % num4 / num5` по шагам.

**Ответ:**

$$4 * 5 = 20$$

$$15 \% 10 = 5$$

$$5 / 5 = 1$$

$$20 + 1 = 21$$

$$12 + 21 = 33$$

**Вопрос 3:**

Чему будет равна переменная `z` после выполнения следующего кода и почему?

```
int x = 8;  
int y = 9;  
int z = x++ + ++y;
```

**Ответ: `z = 18`**

Практическое задание:

## Задача 1

Ваша задача — создать простой калькулятор, который сможет выполнять базовые арифметические операции: сложение, вычитание, умножение, деление, остаток от деления, инкремент, декремент. Калькулятор должен предоставлять пользователю возможность вводить числа и вывод всех математических действий.

Условия выполнения:

Ввод данных:

Пользователь должен вводить два числа (например, целые или дробные).

Операции:

Реализуйте следующие арифметические операции:

Сложение (+)

Вычитание (-)

Умножение (\*)

Деление (/)

Остаток от деления (%)

Инкремент (++)

Декремент (—)

Вывод результата:

После выполнения операции калькулятор должен выводить результат на экран.

ОТВЕТ:

[https://github.com/AntonThere1123/Osnova\\_Algoritma\\_I\\_Programirovaniya\\_by\\_Nosov\\_A.git](https://github.com/AntonThere1123/Osnova_Algoritma_I_Programirovaniya_by_Nosov_A.git)