Колледж электроники и приборостроения

Практическая работа №2

Выполнил: Коломиец Максим

Группа: 23 ИП

**Задание 1**

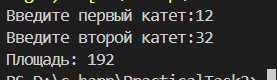
int a;

int b;  
double res;  
Console.Write("Введите первый катет:");

a = int.Parse(Console.ReadLine());  
  
  
Console.Write("Введите второй катет:");

b = int.Parse(Console.ReadLine());  
res = 0.5 a \* b;

Console.WriteLine("Площадь: {0}", res);



**Задание 2**

int number;

int smallestDigit;  
Console.Write("Введите двузначное натуральное число: ");        number = int.Parse(Console.ReadLine());  
        if (number < 10 || number > 99)        {

          Console.WriteLine("Число должно быть двузначным.");

return;

}

        smallestDigit = (number % 10 >= number / 10) ? number / 10 : number % 10;

        Console.WriteLine("Наименьшая цифра в числе {0} равна {1}", number, smallestDigit);



**Задание 3**

 // Задание 3) Число Х является перевертышем числа Y. Зная значение числа Y, найдите исходное число X.  
int x;

int y;

Console.Write("Введите Y:");

y = int.Parse(Console.ReadLine());  
int first;

int second;

int third;  
first = y % 10;

y = y / 10;

second = y % 10;

y = y / 10;

third = y % 10;  
Console.WriteLine("X = {0}{1}{2}", first, second, third);



**Ответы на вопросы контрольные вопросы:**

**1. Из каких разделов состоит программа?**

Программа на C# обычно состоит из следующих разделов:

**Пространства имен (namespaces)**: Используются для организации кода и предотвращения конфликтов имен.

**Классы (classes):** Основные строительные блоки программы, содержащие данные и методы.

**Методы (methods):** Выполняют определенные действия, такие как вычисления, ввод/вывод данных и т.д.

**Переменные (variables):** Хранят данные, с которыми работает программа.

**Операторы (operators):**  Выполняют операции над данными, такие как сложение, вычитание, сравнение и т.д.

**Управляющие конструкции (control structures):** Определяют порядок выполнения кода, такие как циклы и условные операторы.

**2. Зачем нужен оператор присваивания? Какой вид он имеет?**

Оператор присваивания используется для присвоения значения переменной. Он имеет вид:

переменная = выражение;

**переменная**: Имя переменной, которой присваивается значение.

**выражение**: Любое выражение, результатом которого является значение, совместимое с типом переменной.

**3. Что может быть записано в правой части оператора присваивания?**

В правой части оператора присваивания может быть записано любое выражение, результатом которого является значение, совместимое с типом переменной. Это может быть:

**Константа**: Например, `5`, `"Hello"`, `true`.

**Переменная**: Например, `x`, `name`, `isActive`.

**Выражение**: Например, `x + y`, `name.Length`, `isActive && isEnabled`.

**Вызов метода**: Например, `Math.Sqrt(x)`, `Console.ReadLine()`.

**4. Что такое переменная?**

**Переменная** — это именованная область памяти, в которой хранится значение, с которым работает программа. Переменные позволяют:

Хранить данные разных типов.

Изменять значения данных в процессе выполнения программы.

Использовать данные в выражениях и операторах.

**5. Какие правила применяются для создания имен переменных?**

При создании имен переменных в C# необходимо соблюдать следующие правила:

Имена должны начинаться с буквы или символа подчеркивания (\_).

Имена могут содержать буквы, цифры и символы подчеркивания.

Имена не должны совпадать с ключевыми словами языка C#.

Имена должны быть уникальными в пределах своей области видимости.

Имена должны быть понятными и описывать назначение переменной.

Примеры правильных имен переменных:

`age`

`\_name`

`firstName`

`user\_id`

Примеры неправильных имен переменных:

`123variable` (начинается с цифры)

`int` (совпадает с ключевым словом)

`my variable` (содержит пробел)

**Важно**: В C# регистр букв имеет значение. Например, `age` и `Age` — это две разные переменные.