Практическая работа № 3. Условия, циклы, методы, массивы

Теоретическая часть

Условия:

```
string name = "Alex";

if (name == "Tom")
    Console.WriteLine("Вас зовут Tomas");
else if (name == "Bob")
    Console.WriteLine("Вас зовут Robert");
else if (name == "Mike")
    Console.WriteLine("Вас зовут Michael");
else
    Console.WriteLine("Неизвестное имя");
```

Тернарный оператор:

```
[первый операнд - условие] ? [второй операнд] : [третий операнд]
```

Циклы:

Цикл со счетчиком:

```
for ([действия_до_выполнения_цикла]; [условие]; [действия_после_выполнения])
{
    // действия
}
```

Цикл с постусловием:

```
do
{
    действия цикла
}
while (условие)
```

Цикл с предусловием:

```
while (условие)
{
    действия цикла
}
```

Цикл для перебора коллекций:

```
foreach(тип_данных переменная in коллекция)
{
    // действия цикла
}
```

Операторы **continue** и **break** используются в циклах (**break** также используется в конструкции **switch case**).

Операторы continue и break

Иногда возникает ситуация, когда требуется выйти из цикла, не дожидаясь его завершения. В этом случае мы можем воспользоваться оператором break.

Операторы continue и break

Иногда возникает ситуация, когда требуется выйти из цикла, не дожидаясь его завершения. В этом случае мы можем воспользоваться оператором break.

Continue используется для того, чтобы перенестись на новую итерацию цикла, игнорируя весь код, который идет после continue (в рамках цикла).

Break применяется для досрочного выхода из цикла.

Массивы

Примеры объявления массива в C#

```
int[] nums2 = new int[4] { 1, 2, 3, 5 };

int[] nums3 = new int[] { 1, 2, 3, 5 };

int[] nums4 = new[] { 1, 2, 3, 5 };

int[] nums5 = { 1, 2, 3, 5 };
```

Пример перебора массива с помощью цикла

```
int[] numbers = { 1, 2, 3, 4, 5 };
for (int i = 0; i < numbers.Length; i++)
{
    Console.WriteLine(numbers[i]);
}
```

Методы

```
тип_метода имя_метода (тип_параметра1 параметр1, тип_параметра2 параметр2, ...)
{
    // действия метода
}
```

Определение метода

```
void SayHello()
{
    Console.WriteLine("Hello");
}

SayHello(); // Hello
```

```
void PrintMessage(string message)
{
    Console.WriteLine(message);
}

PrintMessage("Hello work");           // Hello work
```

```
string GetMessage()
{
    return "Hello";
}

string message = GetMessage();  // получаем результат метода в переменную message
Console.WriteLine(message);     // Hello
```

Практическая часть

1. Работа с массивом и вычисления

1. Пользователь вводит количество элементов массива.
2. Введите массив из чисел, затем:
    - Найдите минимальный и максимальный элемент массива.
    - Найдите сумму всех элементов массива.
    - Выведите среднее арифметическое элементов массива.

Подсказки:

- Для ввода и обработки элементов массива используйте циклы for или foreach.
- Вычисление минимума, максимума и суммы реализуйте в отдельном методе.

2. Определение простых чисел

1. Пользователь вводит массив чисел.
2. Определите, какие из введённых чисел являются простыми.

3. Выведите список всех простых чисел из массива.

Подсказки:

- Простое число – это число, большее 1, которое делится только на 1 и на само себя.
- Для проверки, простое ли число, реализуйте отдельный метод, который принимает число и возвращает true или false.

3. Массив строк и методы

1. Пользователь вводит массив строк (например, имена).
2. Создайте два метода:
   - Первый метод должен возвращать самую длинную строку из массива.
   - Второй метод должен возвращать количество строк, длина которых больше заданного значения.

Программа должна запросить у пользователя длину, которую нужно использовать для проверки строк во втором методе.

Подсказки:

- Используйте цикл для перебора массива и сравнения длин строк.
- Используйте методы для повышения читаемости и структурирования кода.

4. Подсчет количества элементов массива по условию

1. Пользователь вводит массив целых чисел.
2. Напишите метод, который подсчитывает количество элементов массива, которые больше заданного пользователем значения.
3. Результат работы метода (количество элементов) выводится на экран.

Подсказки:

- Используйте цикл foreach для перебора массива и проверки каждого элемента.
- В методе должен быть один параметр — это число, с которым сравниваются элементы массива.

Задача 5. Подсчет суммы цифр числа

1. Пользователь вводит целое число.

2. Напишите метод, который подсчитывает сумму всех цифр этого числа.
3. Результат работы метода выводится на экран.

Подсказки:

- Для выделения цифр числа используйте деление и остаток от деления на 10.
- Реализуйте метод, который возвращает сумму цифр числа.

Задача 6. Поиск второго максимального элемента в массиве

1. Пользователь вводит массив целых чисел.
2. Напишите метод, который находит второй по величине элемент в массиве.
3. Если в массиве нет второго максимального элемента (например, все элементы одинаковы), выведите соответствующее сообщение.

Подсказки:

- Для нахождения второго максимума сначала найдите максимальный элемент, затем пройдите по массиву и найдите второй по величине элемент.
- Обратите внимание на граничные случаи (например, если массив состоит только из одного элемента).

Задача 7. Проверка палиндрома

1. Пользователь вводит строку.
2. Напишите метод, который проверяет, является ли введенная строка палиндромом (т.е. читается одинаково в обоих направлениях).
3. Программа должна вывести, является ли строка палиндромом или нет.

Подсказки:

- Палиндром — это строка, которая читается одинаково с начала и с конца (например, "level", "radar").
- Игнорируйте регистр символов при проверке, например, "Аба" тоже считается палиндромом.
- Для проверки палиндрома используйте цикл, который сравнивает символы с начала и конца строки.

Задача 8. Удаление дубликатов в массиве

1. Пользователь вводит массив целых чисел.
2. Напишите метод, который удаляет все дублирующиеся элементы из массива и возвращает массив без дубликатов.
3. Выведите новый массив без дубликатов на экран.

   Подсказки:

   - Для решения задачи можно использовать временный список или массив для хранения уникальных элементов.
   - Проверяйте наличие элемента в списке перед добавлением его в результирующий массив.

Задача 9. Транспонирование матрицы

1. Пользователь вводит размерность квадратной матрицы (N x N).
2. Заполните матрицу случайными целыми числами в диапазоне от 1 до 100.
3. Напишите метод, который транспонирует матрицу (меняет местами строки и столбцы) и выводит транспонированную матрицу на экран.

Подсказки:

- Транспонирование матрицы заключается в замене строки на столбец, и наоборот.
- Используйте вложенные циклы для создания и обработки двумерного массива.
- Для генерации случайных чисел используйте класс Random.

Пример

```
Введите размер матрицы: 3
Оригинальная матрица:
12 25 32
43 54 61
78 84 90

Транспонированная матрица:
12 43 78
25 54 84
32 61 90
```