**Методичка C#: Введение (Урок 0)**

**Основные понятия**

- **Алгоритм** – это последовательность действий для достижения поставленной цели. Это план наших действий.

- **Язык программирования** – это язык, который понимает компьютер. Существуют языки, которые понимаем мы (например, русский, английский), и языки программирования, которые понимает компьютер.

- **Программа** – это алгоритм, записанный на определённом языке программирования.

**Этапы решения задач**

1. **Постановка задачи** – описание самой задачи.

2. **Формализация** – перевод задачи на математический язык.

3. **Алгоритмизация** – разработка алгоритма, который решает задачу.

4. **Программирование** – написание кода программы.

5. **Тестирование** – проверка работоспособности программы и корректности выходных данных.

**Среда разработки**

**Среда разработки (IDE)** – это программа, необходимая для написания кода, запуска программы и поиска ошибок. Она удобна, так как включает множество встроенных функций, например, автоматическое заполнение кода, подсветку синтаксиса и указание на ошибки.

Для языка C# рекомендую использовать **Visual Studio**. Это один из лучших инструментов для разработки на C#. Инструкции по установке будут приведены ниже.

\* Установите Visual Studio Code и расширение C# Dev Kit.

**Создание консольного приложения**

1. Откройте терминал в Visual Studio Code (Ctrl + `).

2. Создайте новый проект, выполнив команду:

```bash

dotnet new console -o MyProject

```

Где `MyProject` — это имя вашего проекта.

**Написание первой программы**

```csharp

using System;

namespace FirstProject

{

class Program

{

static void Main(string[] args)

{

// Здесь будем писать код программы

}

}

}

```

Запустите программу с помощью команды:

```bash

dotnet run

```

Вы увидите вывод `Hello, world!` в терминале.

**Разбор кода программы**

- **using System;** – команда, позволяющая использовать библиотеку System, которая содержит полезные функции.

- **Main** – основная функция программы, которая запускается автоматически.

- Комментарии начинаются с `//` и не влияют на выполнение программы.

**\* C# на телефоне**

Для Android существуют приложения, такие как \*\*C# Shell (C# Offline Compiler)\*\*, которые работают без доступа к интернету. Для iOS можно использовать \*\*C# Programming Compiler\*\*.

**Hello, World!**

Традиционно изучение языка программирования начинается с написания программы, выводящей на экран сообщение «Hello, World!»:

```csharp

using System;

namespace HelloWorld

{

class Program

{

static void Main(string[] args)

{

Console.Write("Hello, World!");

}

}

}

```

**Вывод данных**

1. Console.Write

```csharp

static void Main(string[] args)

{

Console.Write("Hello"); // вывод

Console.Write("World!"); // вывод

}

```

Вывод будет: `HelloWorld!`.

1. Console.WriteLine

```csharp

static void Main(string[] args)

{

Console.WriteLine("Hello"); // вывод и перенос на новую строку

Console.WriteLine("World!"); // вывод и перенос на новую строку

}

```

Вывод будет: `Hello World!`

**Форматированный вывод**

```csharp

static void Main(string[] args)

{

Console.WriteLine("5 + 7 = " + 12); // 5 + 7 = 12

Console.WriteLine(5 + " + " + 7 + " = " + 12); // 5 + 7 = 12

Console.WriteLine(3 + " " + 6); // вывод двух чисел через пробел

}

```

**Примечания**

1. Аргументы или параметры команды записываются в круглых скобках.

2. C# различает строчные и заглавные буквы.

3. В конце каждой команды ставится точка с запятой (;).

4. Рекомендуется на одной строке писать только одну команду.

5. Комментарии начинаются с `//`.

6. `Console.WriteLine()` с пустыми скобками вставляет новую пустую строку.