11.12.20г\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

19ИСиП 2Д \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Основы алгоритмизации и программирования

ТЕМА: П\З **Составление программы(ЧАСТЬ 1)**

СОЗДАНИЯ ПРОГРАММЫ

Изначально создание программы можно разделить на следующие этапы:

1. Разработка алгоритма работы программы.
2. Дизайн пользовательского интерфейса. Размеры и свойства окна программы, расположение элементов управление внутри окна (окошки, кнопочки, менюшки и другие) и так далее.
3. Написание логики программы. Это именно та часть, которая реагирует на события от нажатия кнопок, события операционной системы и выводит результат своей работы на экран.

 **П\3 Составление программы**

**Создание приложения**

* [Windows](https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/inside-a-program/hello-world-your-first-program?tabs=windows#tabpanel_CeZOj-G++Q_windows)
* [macOS](https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/inside-a-program/hello-world-your-first-program?tabs=windows#tabpanel_CeZOj-G++Q_macos)

Запустите Visual Studio. В Windows вы увидите следующее изображение:



Выберите **Создать проект** в правом нижнем углу изображения. В Visual Studio отображается диалоговое окно **Новый проект**:



**Примечание**

Если вы запустили Visual Studio впервые, список **Последние шаблоны проектов** будет пустым.

В диалоговом окне "Новый проект" выберите "Консольное приложение (.NET Core)" и нажмите кнопку **Далее**. Присвойте проекту имя, например "HelloWorld", а затем нажмите кнопку **Создать**.

Проект открывается в Visual Studio. Он уже является базовым примером Hello, World! Нажмите клавиши Ctrl + F5 для запуска проекта. Visual Studio выполняет сборку проекта, преобразуя исходный код в исполняемый файл. Затем запускается командное окно, запускающее новое приложение. В окне должен отображаться следующий текст:

КонсольКопировать

Hello World!

C:\Program Files\dotnet\dotnet.exe (process 11964) exited with code 0.

Press any key to close this window . . .

Нажмите любую клавишу, чтобы закрыть окно.

**Элементы программы C#**

Давайте рассмотрим важные части этой программы. Первая строка содержит комментарий. Символы // преобразуют остальную часть строки в комментарий.

C#Копировать

// A Hello World! program in C#.

Вы можете также закомментировать блок текста, заключив его между символами /\* и \*/. Эти действия показаны в следующем примере.

C#Копировать

/\* A "Hello World!" program in C#.

This program displays the string "Hello World!" on the screen. \*/

Консольное приложение C# должно содержать метод Main, в котором начинается и заканчивается управление. В методе Main создаются объекты и выполняются другие методы.

Метод Main является [статическим](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/static) методом, расположенным внутри класса или структуры. В предыдущем примере "Hello World!" он размещается в классе с именем Hello. Вы можете объявить метод Main одним из следующих способов.

* Он может возвращать значение void. Это означает, что программа не возвращает значение.

C#Копировать

static void Main()

{

 //...

}

* Он также может возвращать целое число. Данное целое число — это **код выхода** для приложения.

C#Копировать

static int Main()

{

 //...

 return 0;

}

* С любым из типов возвращаемых значений он может принимать аргументы.

C#Копировать

static void Main(string[] args)

{

 //...

}

-или-

C#Копировать

static int Main(string[] args)

{

 //...

 return 0;

}

Параметр args метода Main является массивом значений типа string, который содержит аргументы командной строки, используемые для вызова программы.

Дополнительные сведения об использовании аргументов командной строки см. в примерах в разделе [Main() и аргументы командной строки](https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/main-and-command-args/).

**Ввод и вывод**

Программы на C#, как правило, используют службы ввода-вывода, предоставляемые библиотекой времени выполнения в .NET. Инструкция System.Console.WriteLine("Hello World!"); использует метод [WriteLine](https://docs.microsoft.com/ru-ru/dotnet/api/system.console.writeline). Это один из методов вывода класса [Console](https://docs.microsoft.com/ru-ru/dotnet/api/system.console) в библиотеке времени выполнения. Он отображает свой строковый параметр в стандартном потоке вывода, за которым следует новая строка. Существуют и другие методы [Console](https://docs.microsoft.com/ru-ru/dotnet/api/system.console) для разных операций ввода и вывода. Если вы добавите в начало программы директиву using System;, классы и методы [System](https://docs.microsoft.com/ru-ru/dotnet/api/system) можно использовать напрямую, не указывая их полные имена. Например, можно вызвать Console.WriteLine вместо System.Console.WriteLine:

C#Копировать

using System;

C#Копировать

Console.WriteLine("Hello World!");

Контрольные вопросы :

1 На каком языке идет написание кода?

2 Как называется программа для создания програмы?

3 Из каких частей состоит составление программы?

Преподователь \_\_\_\_\_\_\_\_\_\_\_\_\_\_Дузаев И.К.