12.12.20г\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Основы алгоритмизации и программирования

ТЕМА: П\З **Составление программы (ЧАСТЬ 2)**

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**

Установка Visual Studio

После загрузки необходимо запустить установщик. В открывшемся окне нам предложат выбрать те компоненты, которые мы хотим установить вместе с Visual Studio. Стоит отметить, что Visual Studio — очень функциональная среда разработки и позволяет разрабатывать приложения с помощью множества языков и платформ.

В нашем случае интересен прежде всего C# и .NET Core, поэтому в наборе рабочих нагрузок можно выбрать только пункт «Кроссплатформенная разработка .NET Core». Можно выбрать и больше опций или вообще все опции, однако стоит учитывать свободный размер на жёстком диске — чем больше опций будет выбрано, тем больше места на диске будет занято.



При инсталляции Visual Studio на ваш компьютер будут установлены все необходимые инструменты для разработки программ, в том числе фреймворк .NET Core. Установка успешна? Начинаем писать программу!

Создание проекта в Visual Studio

Откройте Visual Studio и на стартовом экране выберите пункт «Создание проекта».



На следующем окне в качестве типа проекта нужно выбрать *Консольное приложение (.NET Core)*. Это значит, что мы будем создавать приложение командной строки на языке C#.



В следующем окне зададим название проекта. Пусть будет *HelloWorld*. На этом этапе также можно указать папку, где будет располагаться проект. После этого нажмите кнопку «Создать».

Visual Studio создаст и откроет проект. Окно будет выглядеть так:



В нашем редакторе в центре находится сгенерированный по умолчанию код C#. Впоследствии мы изменим его на свой. Слева находится обозреватель решений, в котором можно увидеть структуру нашего проекта. В данном случае в обозревателе сгенерирована структура по умолчанию. В узле «Зависимости» содержатся сборки, которые добавлены в проект по умолчанию — классы библиотеки .NET, которые будет использовать C#. Однако не всегда все сборки нужны. Лишнее содержимое отсюда потом можно удалить. Или, наоборот, добавить какую-нибудь нужную библиотеку — именно в этом узле она будет размещена.

Hello world

Под узлом «Зависимости» расположен непосредственно сам файл кода программы — Program.cs. Как раз он и открыт в центральном окне. Вначале разберёмся, что весь этот код собой представляет:

**using** System;

**namespace** HelloWorld

{

 **class** Program

 {

 **static** **void** Main(**string**[] args)

 {

 Console.WriteLine("Hello World!");

 }

 }

}

В начале файла мы видим директиву using, после которой идёт название подключаемого пространства имён. Пространства имён необходимы для организации классов в общие блоки. Например, в первой строке подключается пространство имён System, которое содержит фундаментальные и базовые классы платформы .NET.

C# имеет C-подобный синтаксис, и каждая строка завершается точкой с запятой, а каждый блок кода помещается в фигурные скобки. Далее начинается уже наше пространство имён HelloWorld, которое будет создавать отдельную сборку или исполняемую программу. Сначала идёт ключевое слово namespace, после которого следует название пространства имён. По умолчанию Visual Studio даёт ему название проекта. Далее внутри фигурных скобок идёт блок пространства имён.

Пространство имён может включать другие пространства или классы. В нашем случае по умолчанию сгенерирован один класс — Program. Классы объявляются похожим способом: сначала идёт ключевое слово class, а потом название класса, и далее блок самого класса в фигурных скобках.

Класс может содержать различные переменные, методы, свойства, прочие инструкции. В данном случае объявлен один метод Main. В программе на C# метод Main является входной точкой программы, с него начинается всё управление. Это обязательный элемент любой программы.

Слово static указывает, что метод Main статический, а слово void — что он не возвращает никакого значения. Далее в скобках у нас идут параметры метода. string[] args — это массив с именем args, который хранит значения типа string, то есть строки. В данном случае они нам пока не нужны, но в реальной программе это те параметры, которые передаются при запуске программы из консоли.

Внутри метода располагаются действия, которые этот метод выполняет. По умолчанию он содержит одно действие: Console.WriteLine("Hello World!"); - выводит в консоль строку "Hello World!".

Теперь мы можем запустить программу на выполнение с помощью клавиши F5 или с панели инструментов, нажав на зелёную стрелку. И если вы все сделали правильно, то при запуске приложения увидите заветную строку.



Интерактивное приложение на C#

Теперь сделаем всё поинтересней — изменим код на следующий:

**using** System;

**namespace** HelloWorld

{

 **class** Program

 {

 **static** **void** Main(**string**[] args)

 {

 Console.Write("Введите свое имя: ");

 **string** name = Console.ReadLine();

 Console.WriteLine($"Привет {name}");

 }

 }

}

По сравнению с автоматически сгенерированным кодом были внесены несколько изменений. Теперь в методе Main первой строкой выводится приглашение к вводу.

Класс Console, метод которого мы вызываем, находится в пространстве имён System. Это пространство подключено в начале с помощью директивы using. Без подключения пространства имён System невозможно было бы использовать класс Console.

Однако нам необязательно подключать пространство имён. Мы можем даже удалить первую строку, но в этом случае мы тогда должны будем указать полный путь до используемого класса:

System.Console.Write("Введите свое имя: ");

Контрольные вопросы :

1 На каком язые идет написание кода?

2 Как называется программа для создания програмы?

3 Из каких частей состоит составление программы?

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