

# Структура программы и типы данных

Мухортова Н.Н.



# Цель

Цель - получить представление о платформе .NET

Узнать структуру написания программ в среде VisualStudio

Изучить простые типы данных в дополнение к Си



# Возможности, парадигмы языка

Парадигма программирования — это совокупность идей и понятий, определяющих стиль написания компьютерных программ. Это способ концептуализации, определяющий организацию вычислений и структурирование работы, выполняемой компьютером


1. Функциональное программирование
2. Объектно-ориентированное программирование
3. Аспектно-ориентированное программирование
4. Компонентно-ориентированное программирование
5. Событийно-ориентированное программирование



# Функциональное программирование

парадигма программирования, в которой процесс вычисления трактуется как вычисление значений функций в математическом понимании последних (в отличие от функций как подпрограмм в процедурном программировании)

В ФП каждый символ является неизменяемым




# Объектно-ориентированное программирование

Объектно-ориентированное программирование — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования



# Основные принципы ООП

1. абстрагирование для выделения в моделируемом предмете важного для решения конкретной задачи по предмету, в конечном счёте — контекстное понимание предмета, формализуемое в виде класса;
2. инкапсуляция для быстрой и безопасной организации собственно иерархической управляемости: чтобы было достаточно простой команды «что делать», без одновременного уточнения как именно делать, так как это уже другой уровень управления;
3. наследование для быстрой и безопасной организации родственных понятий: чтобы было достаточно на каждом иерархическом шаге учитывать только изменения, не дублируя всё остальное, учтённое на предыдущих шагах;
4. полиморфизм для определения точки, в которой единое управление лучше распараллелить или наоборот — собрать воедино.




# Аспектно-ориентированное программирование

Аспектно-ориентированное программирование — парадигма программирования, основанная на идее разделения функциональности для улучшения разбиения программы на модули.

АОП дополняет объектно-ориентированное программирование, обогащая его другим типом модульности, который позволяет локализовать код реализации crosscutting логики в одном модуле. Такие модули обозначаются термином аспекты, от аспектно-ориентированного программирования.

Аспекты в системе могут изменяться, вставляться, удаляться на этапе компиляции и, более того, повторно использоваться.



# Компонентно-ориентированное программирование

КОП — парадигма программирования, существенным образом опирающаяся на понятие компонента — независимого модуля исходного кода программы, предназначенного для повторного использования и развёртывания и реализующегося в виде множества языковых конструкций (например, «классов» в объектно-ориентированных языках программирования), объединённых по общему признаку и организованных в соответствии с определёнными правилами и ограничениями

Компонентно-ориентированный подход появился в 1987 году[1], когда Никлаус Вирт предложил для языка «Оберон» паттерн написания блоков. Данный паттерн сформировался при изучении проблемы «хрупких» базовых классов, возникающей при построении объёмной иерархии классов.






# Паттерн

Шаблон проектирования или паттерн в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

Паттерн заключался в том, что компонент компилируется отдельно от других, а на стадии выполнения — необходимые компоненты подключаются динамически



# Событийно-ориентированное программирование

Событийно-ориентированное программирование — парадигма программирования, в которой выполнение программы определяется событиями — действиями пользователя (клавиатура, мышь), сообщениями других программ и потоков, событиями операционной системы

СОП можно также определить как способ построения компьютерной программы, при котором в коде (как правило, в головной функции программы) явным образом выделяется главный цикл приложения, тело которого состоит из двух частей: выборки события и обработки события



# Платформа .NET

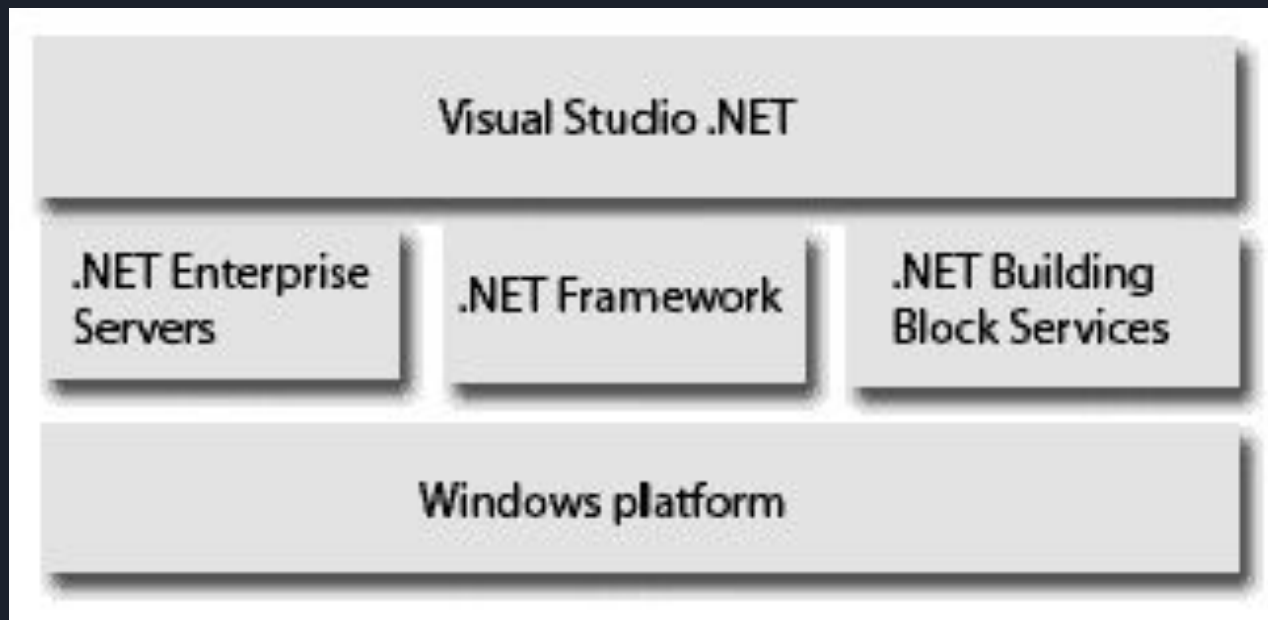
.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime, CLR

# Схема выполнения программы





# Структура платформы .NET





# Visual Studio .NET

Среда разработки, делающая разработку приложений быстрой и приятной.

В Visual Studio .NET входят средства разработки на языках C# ( произносится "Си шарп" ), Visual Basic, Visual C++, а также ряд дополнительных технологий, облегчающих дизайн, коллективную разработку и внедрение Windows-приложений, web-приложений и web-сервисов.



# .NET Enterprise Servers

набор серверных продуктов, сокращающих время, требуемое для разработки крупномасштабных бизнес-систем.

Сюда входят:

1. Сервер баз данных SQL Server 2000, а также широкий ряд других серверных приложений Application Center 2000,
2. BizTalk Server 2000, Commerce Server 2000,
3. Exchange Server 2000,
4. Host Integration Server,
5. Internet Security Acceleration Server 2000.



# .NET Framework

сердце платформы .NET

включает в себя общеязыковую среду выполнения Common Language Runtime ( CLR ) и общую структуру классов, которые могут использоваться всеми языками .NET.

Среди библиотек классов .NET Framework есть практически все для разработки 90% типичных бизнес-приложений. Классы для разработки графических Windows- и web- приложений, классы для работы с базами данных и XML, и многое другое





# .NET Building Block Services

Microsoft Passport, позволяющий установить единое имя пользователя и пароль на всех сайтах. Или такие сервисы, как календарь, поисковый сервис.

Сервисы могут быть платными.



# Windows platform

Самый нижний уровень, на котором работает все вышеперечисленное. Это операционная система, которая может быть одной из нескольких Windows-платформ, включая Windows XP, Windows 2000, Windows 7, Windows 10.



# Типы проектов

- - Windows-приложение использует элементы интерфейса Windows, включая формы, кнопки, флажки и пр.;
- - консольное приложение выполняет вывод на экран «на консоль», то есть в окно командного процессора;
- - библиотека классов объединяет классы, которые предназначены для использования в других приложениях;
- - веб-приложение – это приложение, доступ к которому выполняется через браузер (например, Internet Explorer) и которое по запросу формирует веб-страницу и отправляет её клиенту по сети;
- - веб-сервис – компонент, методы которого могут вызываться через Интернет



# Программа “Hello, World!”

```
using System;

namespace ConsoleApplication1{

    class Class1{

        static void Main(string[ ] args){

            Console.WriteLine(“Hello, World!”);

        }

    }

}
```



# Простые типы данных

byte: хранит целое число от 0 до 255 и занимает 1 байт. Представлен системным типом `System.Byte`

sbyte: хранит целое число от -128 до 127 и занимает 1 байт. Представлен системным типом `System.SByte`

short: хранит целое число от -32768 до 32767 и занимает 2 байта. Представлен системным типом `System.Int16`

ushort: хранит целое число от 0 до 65535 и занимает 2 байта. Представлен системным типом `System.UInt16`



# Простые типы данных

`decimal`: хранит десятичное дробное число. Если употребляется без десятичной запятой, имеет значение от 0 до  $\pm 79\,228\,162\,514\,264\,337\,593\,543\,950\,335$ ; если с запятой, то от 0 до  $\pm 7,9228162514264337593543950335$  с 28 разрядами после запятой и занимает 16 байт. Представлен системным типом `System.Decimal`

`object`: может хранить значение любого типа данных и занимает 4 байта на 32-разрядной платформе и 8 байт на 64-разрядной платформе. Представлен системным типом `System.Object`, который является базовым для всех других типов и классов .NET



# Использование суффиксов

`float a = 3.14F;`

`float b = 30.6f;`

`decimal c = 1005.8M;`

`decimal d = 334.8m;`



# Неявная типизация

```
var hello = "Hello to World";
```

```
var c = 20;
```

```
//переменную необходимо сразу инициализировать
```

```
Console.WriteLine(c.GetType().ToString());
```

```
Console.WriteLine(hello.GetType().ToString());
```





# Кортежи

Кортеж представляет набор значений, заключенных в круглые скобки

```
var tuple = (5, 10);    или    (int, int) tuple = (5, 10);
```

```
Console.WriteLine(tuple.Item1);    // 5
```

```
(string, int, double) person = ("Tom", 25, 81.23);
```

Мы также можем дать названия полям кортежа

```
var tuple = (count:5, sum:10);
```



# Использование кортежей

```
static void Main(string[] args){  
  
    var tuple = GetValues();  
  
    Console.WriteLine(tuple.Item1); // 1  
  
    Console.WriteLine(tuple.Item2); // 3  
  
    Console.Read();  
  
}
```



# Выводы

Платформа .NET обеспечивает кроссплатформенность

C# поддерживает дополнительные простые типы данных, которых нет в C.

C# полностью объектно-ориентированный язык, поэтому изучение его начнется со знакомства с ООП

C# поддерживает все современные парадигмы программирования