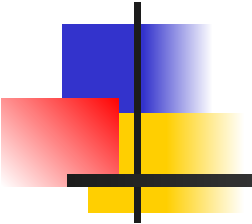


Модуль 1

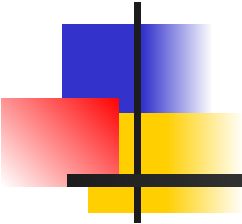
Начало работы с C#



Обзор модуля

В этом модуле вы изучите:

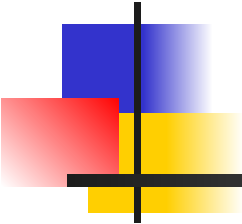
- Введение в C#
- Особенности языка
- Среду Visual Studio 2005
- Элементы Microsoft Visual Studio 2005 IDE



Занятие 1 - Введение в C#

На этом первом занятии, **Введение в C#**, вы научитесь:

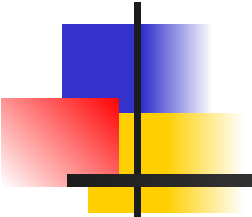
- Формулировать ловушки C/C++.
- Определять C# как новый язык.
- Описывать цель языка C#.
- Описывать .NET Framework.
- Перечислять и объяснять компоненты .NET Framework.
- Описывать архитектуру .NET Framework.
- Перечислять другие компоненты .NET Framework.
- Объяснять общезыковую исполняющую среду (CLR) и Microsoft Intermediate Language (MSIL).



Ловушки C/C++

Некоторые недостатки:

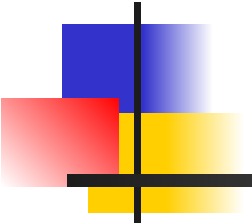
- Сложен в изучении для начинающих
- Нельзя проверить код до компиляции
- Сложность отладки без дорогих отладочных инструментов
- Требуется много времени на разработку
- Сложное подключение в базах данных
- Проблемы при модификации данных
- Сложность при реализации
- Не содержит никаких структур



Необходимость нового языка

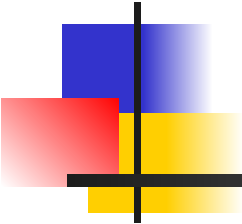
C# был разработан для предоставления следующих преимуществ:

- Создания очень простых и мощных инструментов для разработки взаимодействующих, масштабируемых и надежных приложений
- Создания полностью объектно-ориентированной архитектуры
- Поддержки мощной компонентно-ориентированной разработки
- Получения доступа ко многим возможностям, доступным только в C++, при сохранении простого в использовании быстрого инструмента разработки, например Visual Basic
- Обеспечения легкого знакомства с языком программистов, переходящих с C или C++
- Написания приложений для настольных ПК и мобильных устройств



Цели языка C#

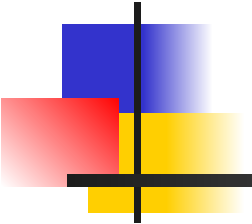
- Microsoft.NET ранее известен как Next Generation Windows Services (NGWS).
- Это абсолютно новая платформа для разработки нового поколения Windows/Web приложений.
- C# это объектно-ориентированный язык производный от C и C++.
- Представляет собой простой, эффективный, производительный, объектно-ориентированный язык.



Основы .NET Framework

.NET Framework разработан для:

- Обеспечения единообразного объектно-ориентированного программного окружения
- Минимизации конфликтов при развертывании приложений и управлении версиями за счет предоставления выполняющего код окружения
- Обеспечения безопасного выполнения кода за счет предоставления выполняющего код окружения
- Предоставления последовательного опыта разработки через разные типы приложений, например Windows-приложения и Web-приложения

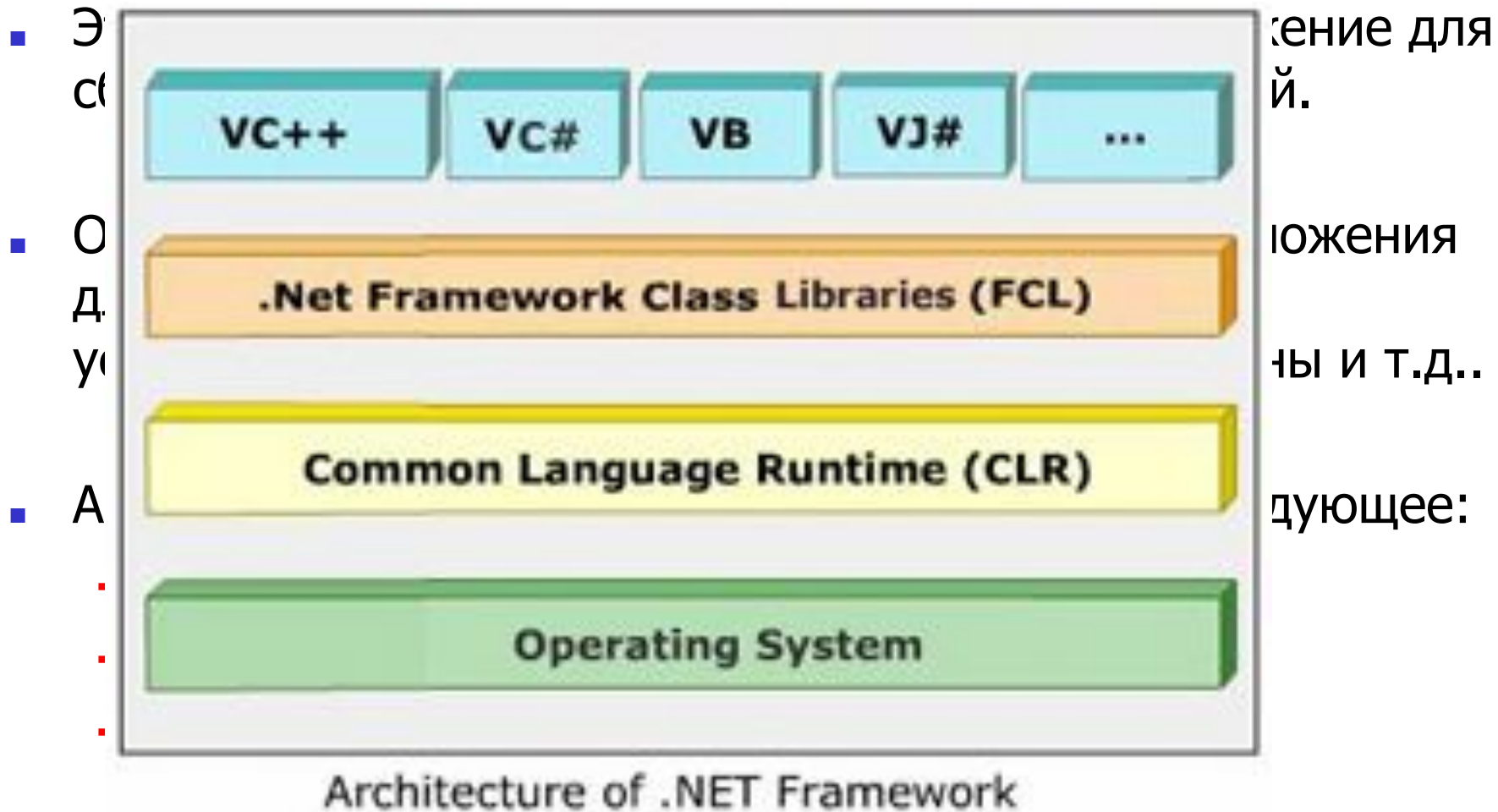


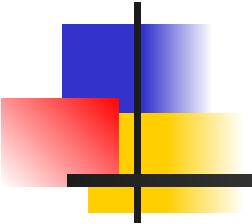
Компоненты .NET Framework

Библиотека классов .NET Framework содержит следующие компоненты:

- это всеобъемлющая объектно-ориентированная коллекция повторно используемых типов (FCL)
- используется для разработки приложений с графическим интерфейсом пользователя (GUI), которые могут использоваться в Web.
 - выполняет различные функции, например:
 - Управление памятью
 - Выполнение кода
 - Обработка ошибок
 - Проверка кода на безопасность
 - Сборка мусора

Организация .NET Framework

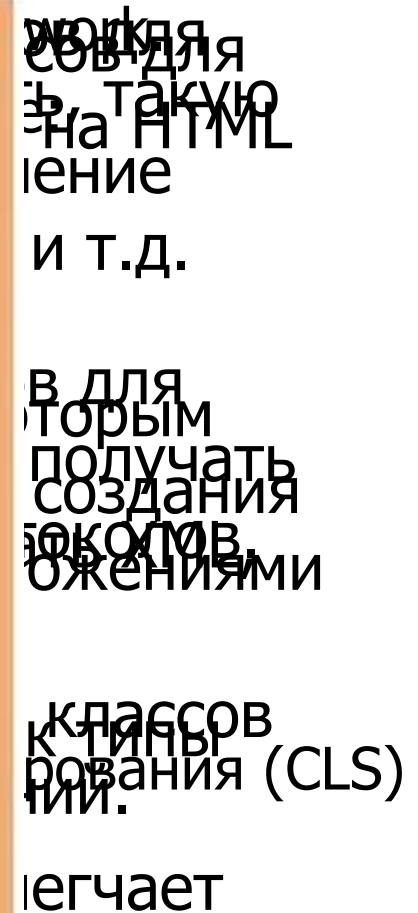




Использование .NET Framework

- Программисты, разрабатывающие приложения, используют один из языков, поддерживаемых .NET.
- Эти приложения используют базовые библиотеки классов предоставленные .NET FCL.
- Следующая команда может быть использована для вывода текстового сообщения на экран:

```
System.Console.WriteLine(".NET Architecture");
```

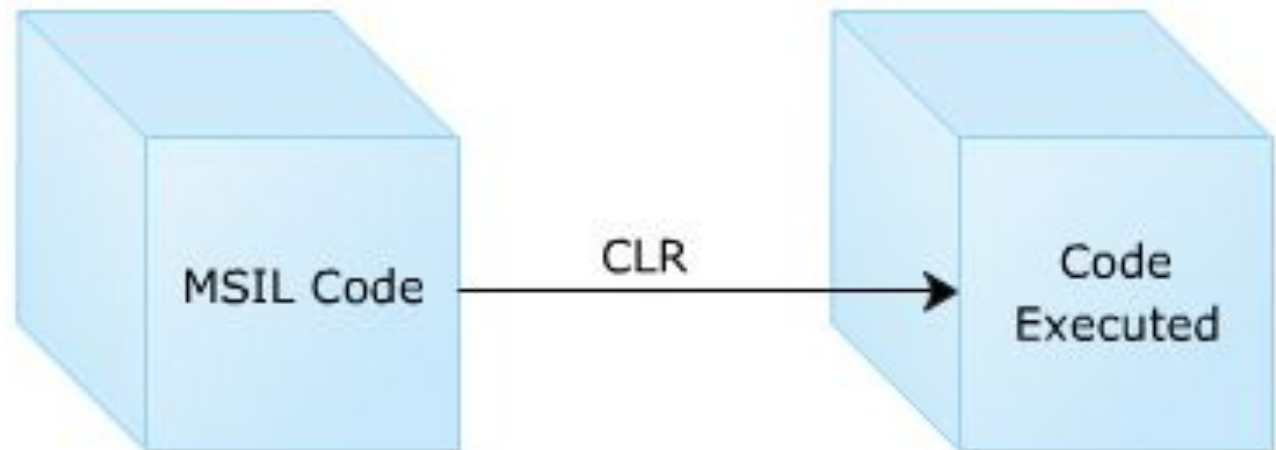


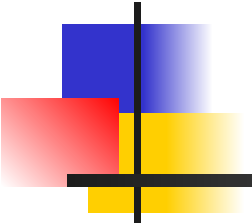
- _____



Общезыковая исполняющая среда (CLR)

- CLR это виртуальная машина компонентов .NET, которая используется для преобразования кода MSIL в машинные инструкции.
- Это происходит во время работы компилятора Just-In-Time (JIT), присутствующего в CLR.

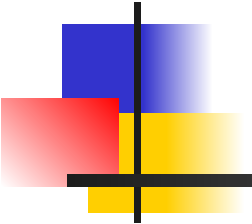




Занятие 2 – Возможности языка

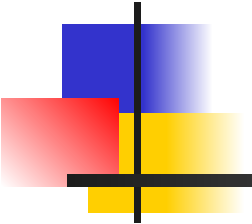
На втором занятии, **Возможности языка**, вы научитесь:

- Перечислять базовые возможности C#.
- Определять общие приложения C#.
- Формулировать преимущества C#.
- Определять управление памятью и сборщик мусора.



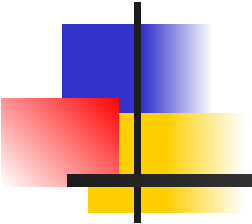
Базовые возможности C#

- **Объектно-ориентированное программирование**
C# это язык программирования, созданный для сосредотачивается на объектах, таким образом, разработки широкого спектра приложений, написанный однажды код может быть повторно выполняемых в .NET Framework.
- **Проверка безопасности типов**
Неинициализированные переменные не могут быть использованы.
■ Некоторые ключевые особенности.
- **Сборщик мусора**
Объектно-ориентированное программирование выполняет автоматическое управление памятью и экономит время программиста.
- **Стандартизация европейской ассоциацией изготовителей компьютеров (ЕСМА)**
Стандартизация европейской ассоциацией изготовителей компьютеров (ЕСМА) - заданный синтаксис и ограничения используются для создания стандартных программ C#.
- **Обобщенные методы и типы**
Обобщенные методы и типы поддерживают код, который остается неизменным, но типы данных параметров могут менять во время использования.



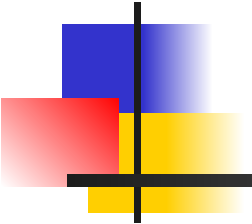
Приложения C#

- C# может использоваться в различных приложениях, например таких как:
 - Игры
 - Крупные промышленные приложения
 - Мобильные приложения для карманных персональных компьютеров (PC), цифровых секретарей(PDA) и сотовых телефонов
 - Простые изолированные настольные приложения, например система управления библиотекой, генератор сводной экзаменационной ведомости студентов и другие
 - Комплексные распределенные приложения, которые охватывают города или целые страны



Преимущества C#

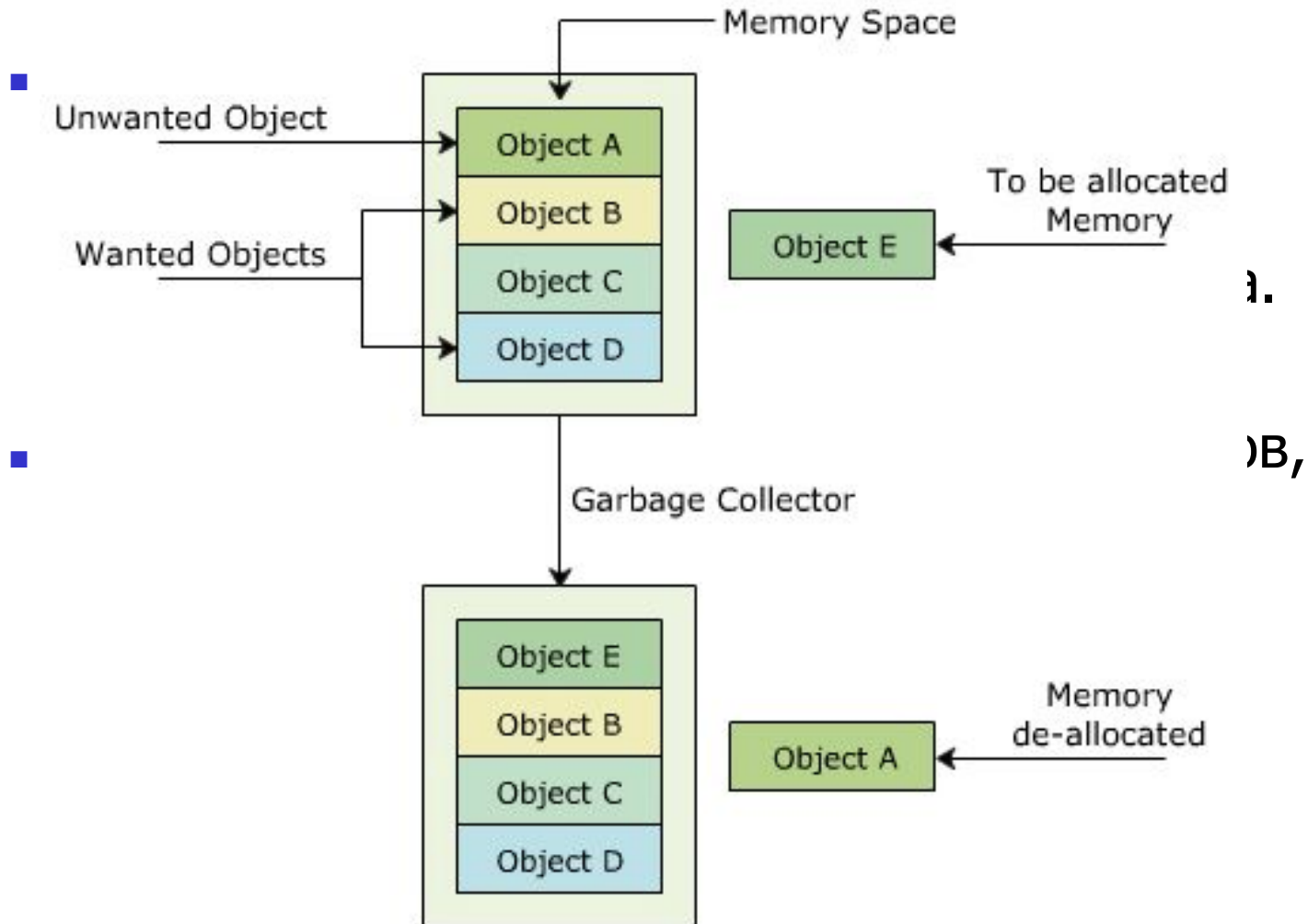
- **Поддержка нескольких языков** – Код C# предпочтительнее, чем C++, поскольку он написанный на любом языке .NET, может быть проще и удобнее в использовании.
- **Общие протоколы Интернет** - .NET предлагает все преимущества поддержки XML, который является предпочтительным выбором для форматирования информации в Интернет.
- **Простое развертывание** – Развертывание приложения C# упрощено благодаря концепции сборки.
- **Документация XML** - Комментарии могут быть переведены в формат XML и затем использованы для документирования кода.

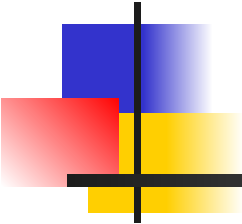


Управление памятью

- В таких языках программирования как С и С++, выделение и освобождение памяти выполняется вручную.
- С# предоставляет возможность выделять и освобождать память, используя автоматическое управление памятью.
- Автоматическое управление памятью улучшает качество кода, повышает производительность и продуктивность.

Сборщик мусора

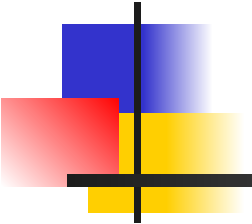




Занятие 3 - Среда Visual Studio 2005

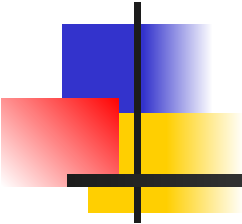
На третьем занятии, **Среда Visual Studio 2005**, вы научитесь:

- Описывать продукт Microsoft Visual Studio 2005.
- Перечислять различные редакции Visual Studio 2005.
- Перечислять языки, поддерживаемые Visual Studio 2005.
- Перечислять и объяснять возможности Visual Studio 2005.



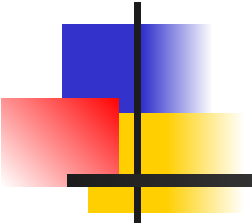
Введение в Visual Studio 2005

- Это полный набор средств разработки для создания высокопроизводительных настольных приложений, XML Web-сервисов, мобильных приложений и приложений ASP Web.
- Также используется для упрощения командного проектирования, разработки и развертывания промышленных решений.
- Основные преимущества:
 - Увеличивает продуктивность разработчика
 - Разработка приложений для Microsoft .NET Framework 2.0
 - Разработка приложений для портативных устройств использующих Microsoft .NET Compact Framework 2.0



Редакции Visual Studio 2005

- **Team System Edition** - используется для разработки и тестирования приложений, созданных с помощью Visual Studio Team System. Эта редакция предназначена для разработчиков, работающих в команде. Она поддерживает все функции, которые есть в других редакциях, а также добавляет функции для работы с командой, такие как управление версиями, отслеживание изменений и т.д.
- **Express Edition** - используется для разработки и тестирования приложений, созданных с помощью Visual Studio Express Edition. Эта редакция предназначена для разработчиков, работающих в одиночку. Она поддерживает все функции, которые есть в других редакциях, а также добавляет функции для работы с командой, такие как управление версиями, отслеживание изменений и т.д.
- **Professional Edition** - используется разработчиками, работающими в одиночку или в небольшой команде. Она поддерживает все функции, которые есть в других редакциях, а также добавляет функции для работы с командой, такие как управление версиями, отслеживание изменений и т.д.

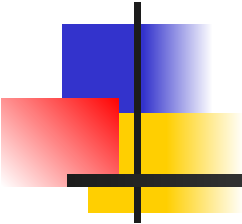


Языки в Visual Studio 2005

- Visual Studio 2005 поддерживает несколько языков программирования.
- Языки, поддерживаемые Visual Studio 2005:
 - Visual Basic
 - Visual C++
 - Visual C#
 - Visual J#

Возможности Visual Studio 2005

- **Шаблон кода** - Небольшие модули исходного кода C#, которые разработчик может быстро использовать с помощью горячих клавиш.
- **Всеобъемлющая инструментальная платформа** - Visual Studio 2005 предоставляет новые возможности, например:
 - **Автоматическое восстановление** - индивидуальное автоматическое регулярное сохранение платформы.
 - **Снижение сложности разработки** - Позволяет снижать сложность разработки, предоставляя клиентам более простые решения.
 - **Сохранение и восстановление** - NET Framework, Visual Studio 2005 предоставляет возможность сохранения и восстановления приложений.
- **Заметки для редактирования** - Представляет собой интеллектуальное восприятие кода, позволяющее разработчику сохранять и восстанавливать код, который он вводит, в виде заметки, которую можно сохранить на диск.



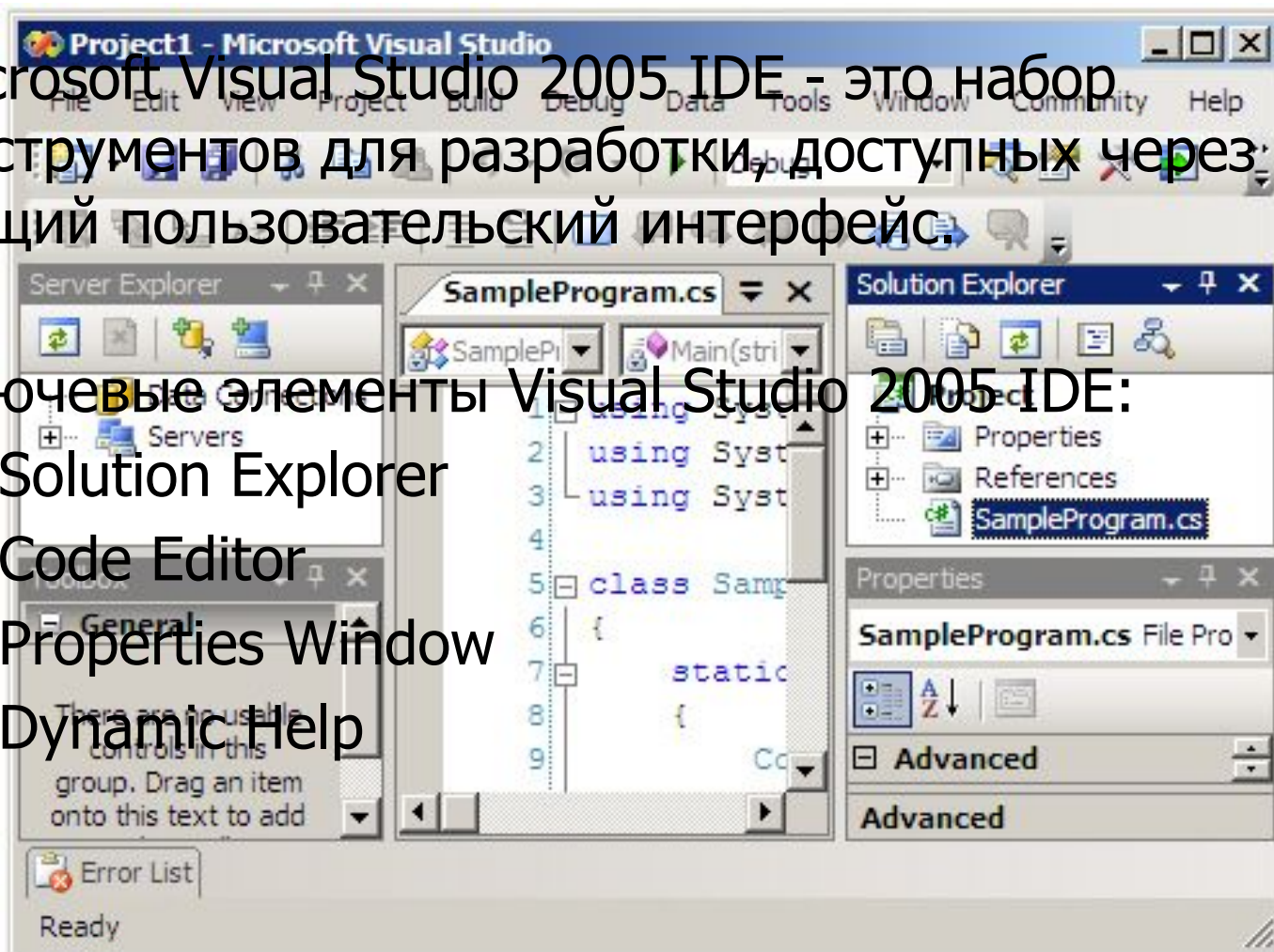
Занятие 4 - Элементы Microsoft Visual Studio 2005 IDE

На последнем занятии, **Элементы Microsoft Visual Studio 2005 IDE**, вы научитесь:

- Узнавать ключевые элементы Visual Studio 2005 IDE.
- Описывать процесс компиляции и выполнения программы C# (из командной строки и из IDE).

Ключевые элементы

- Microsoft Visual Studio 2005 IDE - это набор инструментов для разработки, доступных через единый пользовательский интерфейс.
- Ключевые элементы Visual Studio 2005 IDE:
- Solution Explorer
- Code Editor
- Properties Window
- Dynamic Help



Команда "csc"

Используется для компиляции C# программы
Команда компилятора C# (C Sharp Compiler)

Синтаксис

`D:\Source Code\SampleProgram>csc SampleProgram.cs`

`D:\Source Code\SampleProgram>SampleProgram`

где,

`D:\Source Code\SampleProgram` путь к программе.

`csc SampleProgram.cs`

где,

`SampleProgram`: имя программы для компиляции.

`This is sample C# program`

`D:\Source Code\SampleProgram>`

Сборка и выполнение

Компиляция C# программы

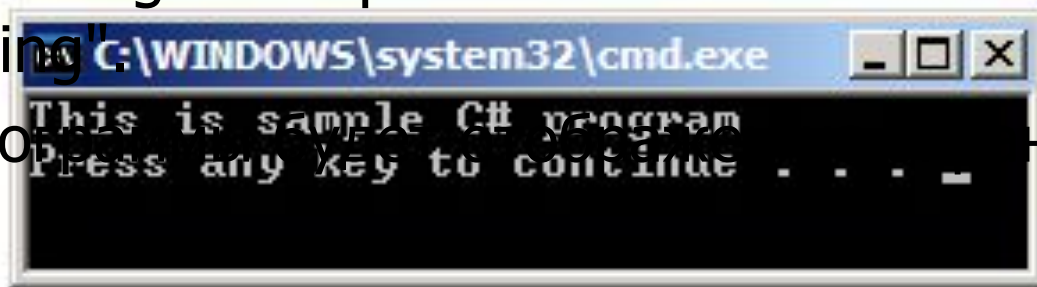
Выберите "Build <имя приложения>" из меню "Build".

Это действие создаст исполняемый файл (.exe).

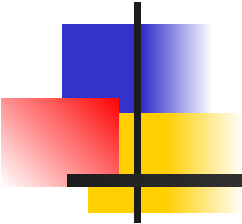
Выполнение программы

В меню "Debug" выберите "Start Without Debugging".

Вывод программы:



```
C:\WINDOWS\system32\cmd.exe
This is sample C# program
Press any key to continue . . .
```



Заключение

- **Введение в C#**

- C# это объектно-ориентированный язык производный от C и C++.

- **Особенности языка**

- Ключевая особенность – это его объектно-ориентированность.
- Он также поддерживает такие возможности, как проверка безопасности типов, сборка мусора, ECMA стандартизация и обобщения.

- **Среда Visual Studio 2005**

- Это полный набор средств разработки для создания высокопроизводительных настольных приложений, XML Web-сервисов, мобильных приложений и приложений ASP Web.

- **Элементы Microsoft Visual Studio 2005 IDE**

- Это набор инструментов разработки, доступных через общий пользовательский интерфейс.