



Высшая школа информационных
технологий и информационных систем



InformatiCS

План лекции 1

1.1 Введение в алгоритмизацию

- Понятие алгоритма и компьютерной программы
- Разрешимые и неразрешимые задачи, алгоритмическая сложность
- Структура компьютера по фон Нейману
- Языки низкого и высокого уровня
- Компилируемые и интерпретируемые языки (примеры, различия)

1.2 Технологии Microsoft .NET NET Framework. История, идеология

- CLR, CTS, CLS, MSIL
- Языки .NET. Межъязыковое взаимодействие
- Язык C#

Этапы компиляции и исполнения программ в .NET Framework на примере языка C#

Алгоритмизация и программирование

- Задача
- Алгоритм
- Программа

Каковы отношения между этими понятиями?

Алгоритмическая разрешимость

Как соотносятся множества
задач,
алгоритмов,
программ?

Все ли задачи можно решить?

Сложность алгоритмов

Как оценивать сложность алгоритмов?

Для чего нужна эта оценка?

Как влияет на разработку программ?

Компьютерные программы

- Что для нас программы?
- Как представляются в компьютере?
- Как выполняются и что используют?

Архитектура компьютера

- Компьютер обрабатывает данные и инструкции (программы)
- Как хранятся и обрабатываются программы и данные?

Фон Неймановская архитектура

- Совместное хранение данных и инструкций
- Использование бинарного представления
- ! Недоступность данных и инструкций одновременно
- ! Производительность упирается в шину между процессором и памятью

Гарвардская архитектура

- Раздельное хранение данных и инструкций на разных физических устройствах
- Шины доступа к данным тоже разделены
- Одновременная работа с данными и инструкциями
- ! Высокая стоимость
- ! У процессора вдвое больше выходов
- ! Непересекающиеся шины

Компромисс

- Гибридные архитектуры
- Общая шина доступа к внешним данным, внутри процессора – шина данных и шина команд
- Кэширование, разгружающее шины и ускоряющее работу

Представление информации

- Бинарное представление
- С какими данными работают программы?

Типы данных

Примитивные данные:

- Числа
 - Целые
 - Вещественные
- Символы
- Строки
- Логические значения

Операции:

Составные типы данных

Данные могут логически связываться в структуры данных

- Однотипные – массивы
- Разнотипные – структуры, классы

Структуры данных и алгоритмы

Алгоритмы + Структуры данных = Программы
(Н. Вирт, 1976)

- Структура данных – логическое связывание данных
- Реализация абстрактных типов данных
- Объекты в ООП

Разработка программ

Языки программирования

- Низкого уровня
- Высокого уровня

Подходы к проектированию

- Восходящий
- Нисходящий

Представление алгоритмов

- Последовательное исполнение
- Условный и безусловный переход
- ! Спагетти код
- ! Сложно понять смысл, сложно поддерживать, масштабировать

Структурные алгоритмы

- Последовательное исполнение
- Ветвление
- Цикл
- Блок-схемы (потокосные диаграммы)
- Теорема Бёма-Якопини (1966)

Структурные алгоритмы

Задачи и требования к инструментам

- Большие размеры задач
- Необходимость в масштабируемости
- Необходимость в простоте, высокоуровневом описании
- Необходимость поддержки командной разработки
- Необходимы средства повторного использования кода
- Необходима модульность
- Необходим набор типовых решений

Языки программирования

- **Вызов – высокие требования к инструментам разработки программ**
- **Ответ – современные языки и среды разработки (IDE)**
- **Среди них – C# и MS Visual Studio**

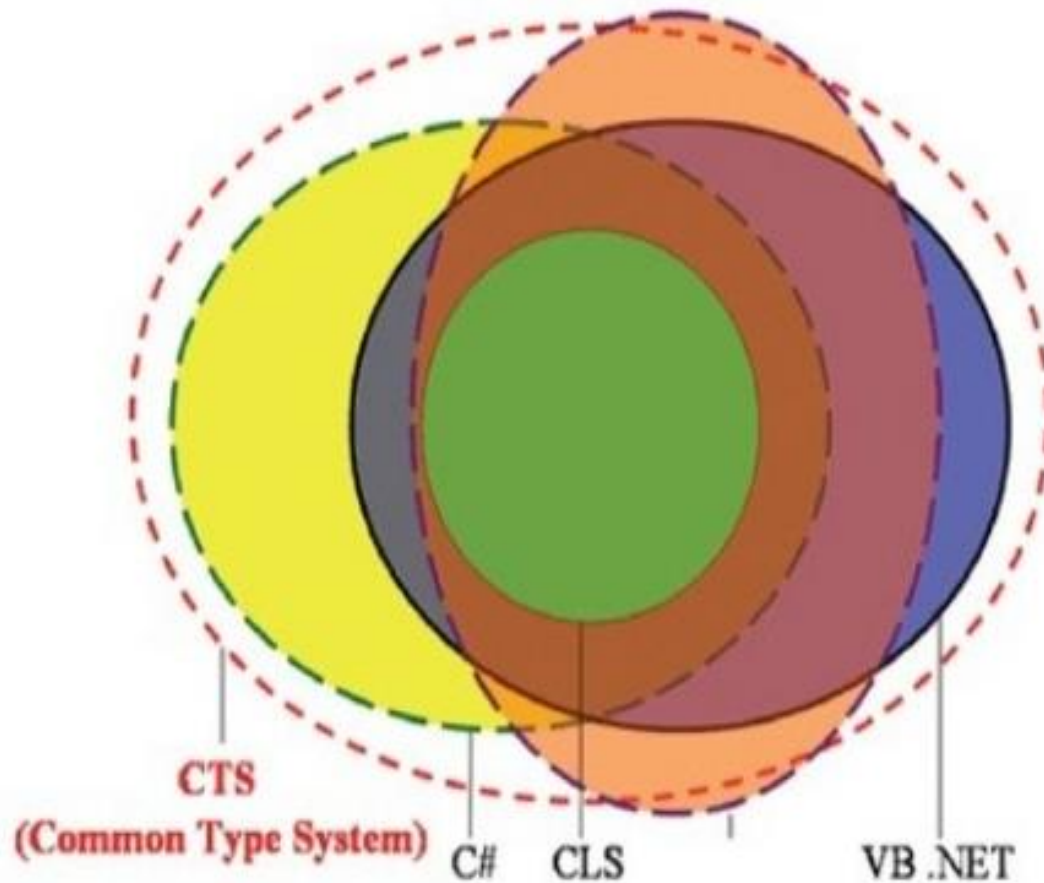
Что такое .NET Framework

- Программная платформа компании Microsoft
- Содержит инструменты и библиотеки для быстрой и удобной разработки ПО
- Должна быть установлена на компьютере для исполнения .NET приложений

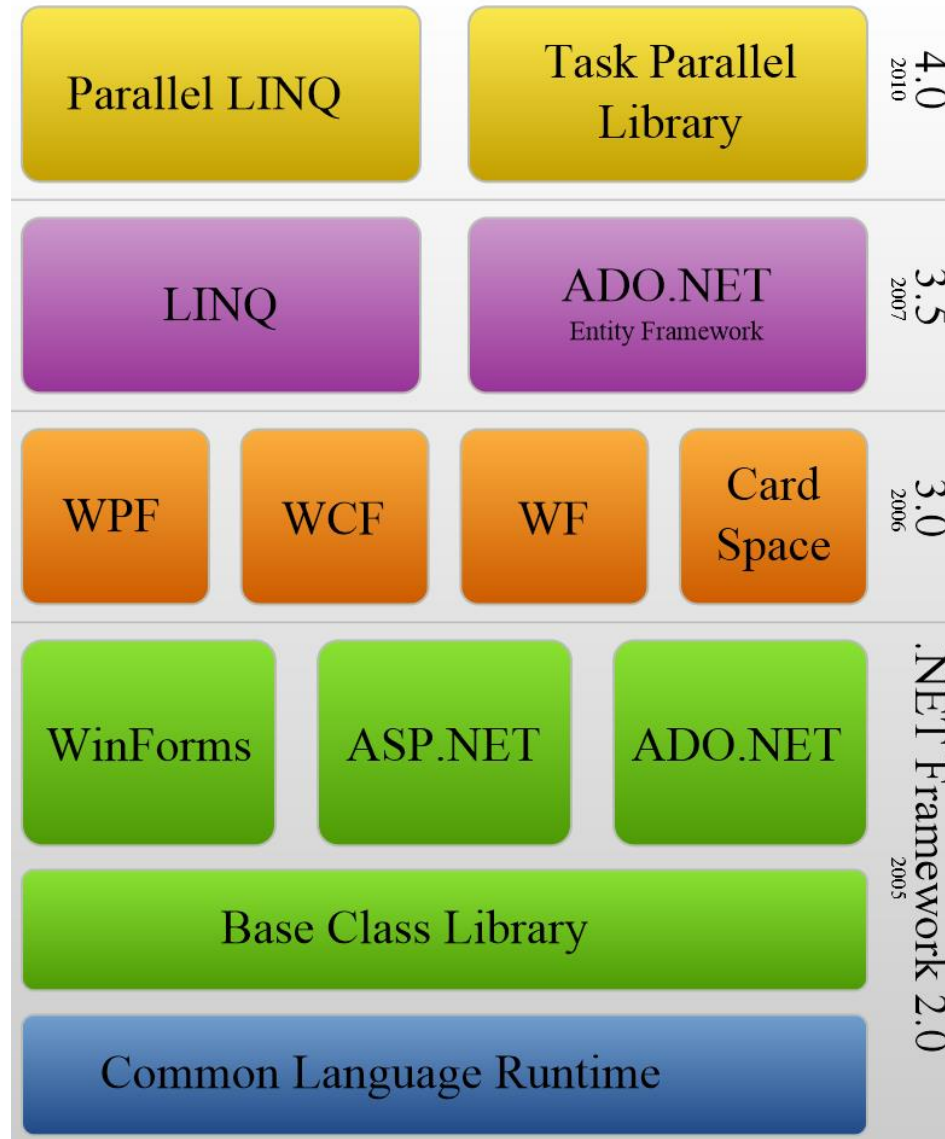
Архитектура .NET Framework



CLS, .NET языки, CTS

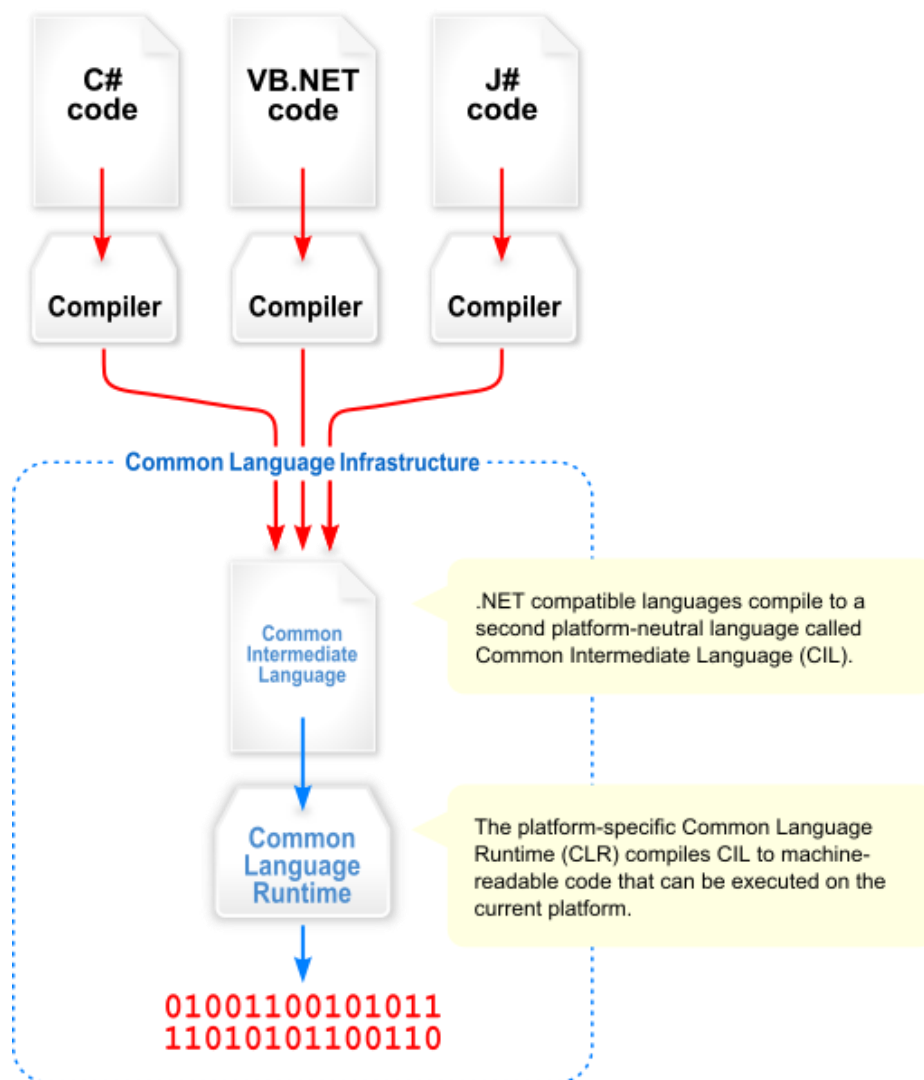


Библиотека классов



Общезыковая среда исполнения

- Загружает и исполняет код
- Компилирует код на промежуточном языке в машинный код
- Разделяет процессы и память
- Управляет памятью
- Следит за безопасностью кода и доступа
- Связывает модули управляемого кода



Средства разработки .NET



Microsoft Visual Studio

<https://www.visualstudio.com/products/visual-studio-community-vs>

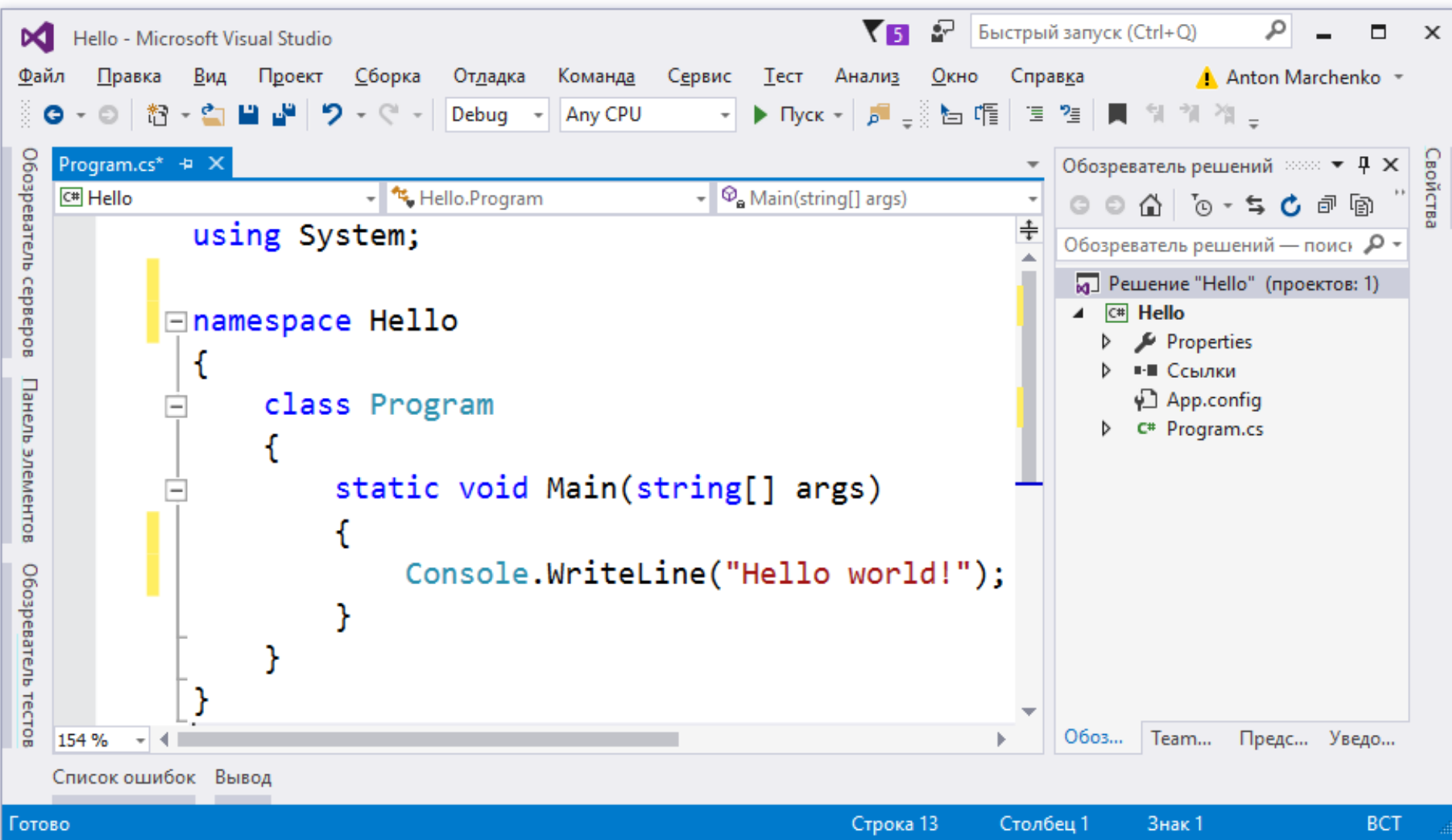
Языки программирования C#

- Один из самых популярных ЯП
- Прост в обучении и использовании
- Современный
- Универсальный
- Мультипарадигменный
- Типобезопасный

Использование языка C#

- Разработка десктоп, мобильных приложений (универсальных приложений платформы Windows [UWP])
- Кроссплатформенная разработка приложений
- Разработка веб-приложений ASP.NET MVC
- Работа с базами данных
- Разработка сервисов (WPF)

Hello world 😊



Вопросы?

Материалы курса:

<http://amantix.ru/InformatiCS/>

Контактные данные: marchenko@it.kfu.ru