

Университет машиностроения

Кафедра «Автоматика и процессы управления»

Дисциплина

Информационные технологии

2 семестр

Тема 14

Основы ООП

**Введение в объектно-ориентированное
программирование**

Список рекомендуемых сайтов:

1. Руководство по программированию на C# от Microsoft
<http://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
2. Википедия (<http://ru.wikipedia.org>)
3. Викиучебник (<http://ru.wikibook.org>) раздел «Информационные технологии»
4. Электронная библиотека (<http://wmate.ru/ebooks/>)
5. И другие сайты по тематике курса.

Предпосылки возникновения ООП и КСИ

Концепция объектно-ориентированного программирования является средством решения проблемы **разработки систем средней сложности** и с целью обеспечения **возможности дистрибуции компонент** программного обеспечения

«Приятные особенности» систем средней сложности:

1. Невозможность единовременного обзора системы;
2. Потребность в групповой разработке из-за высокой общей трудоёмкости (и конкуренции);
3. Необходимость надежного инструмента обеспечения повторного использования кода (в том числе чужого);
4. Ограниченность выразительных способностей традиционных языков программирования для логической изоляции кода алгоритма от остальных частей программы;
5. Проблемы анализа надежности работы...

Предпосылки...

Развитие средств доставки приложений



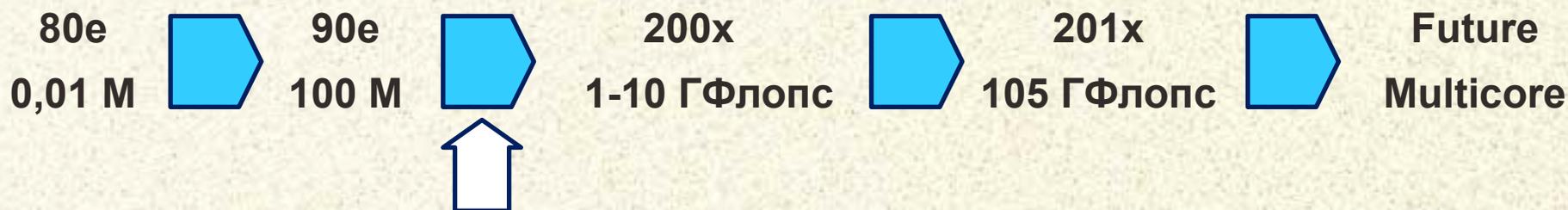
Развитие каналов связи



Рост размера и функциональности приложений



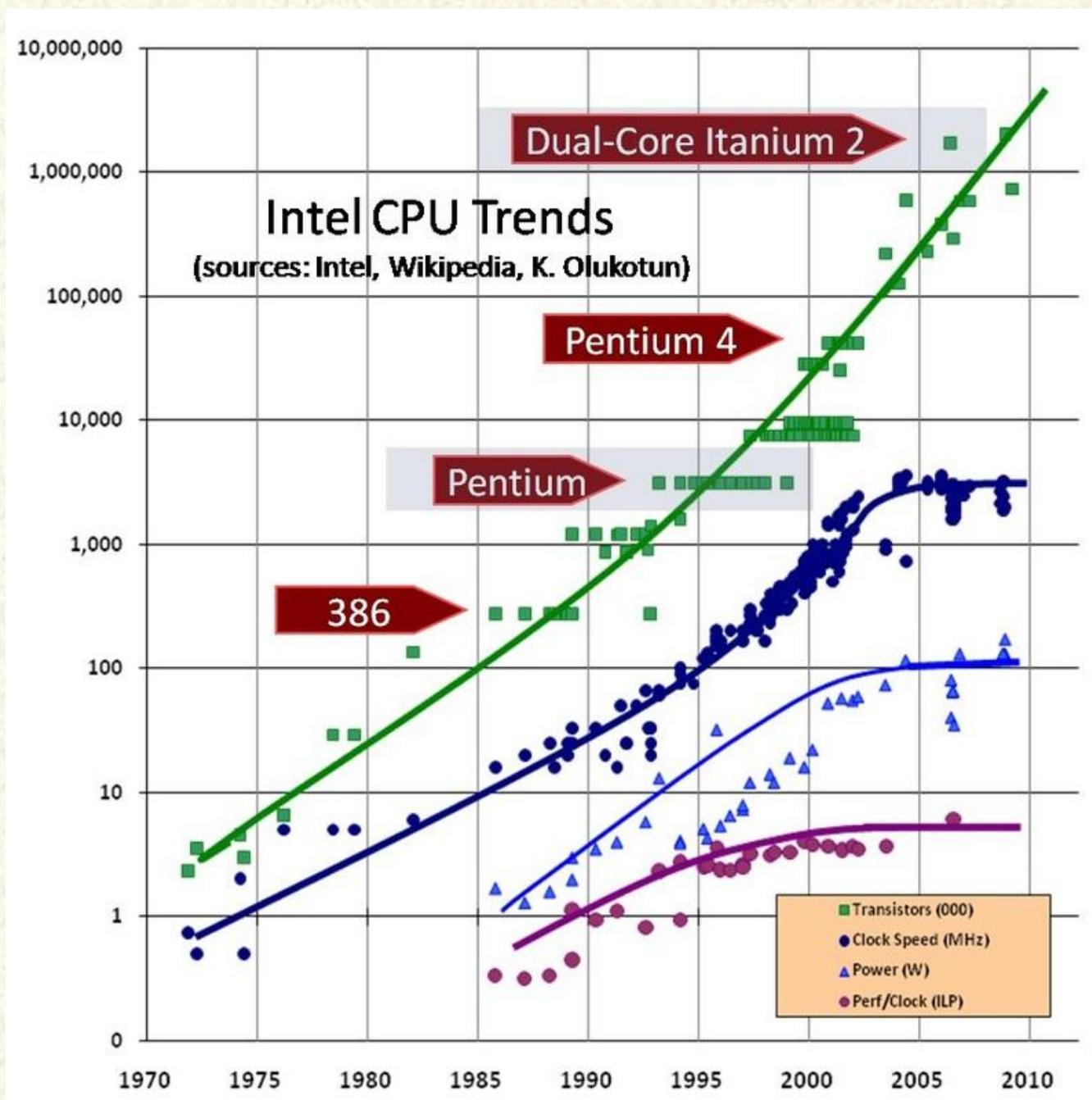
Рост возможностей вычислительных систем (Flops)



Концепция Rich Internet Application

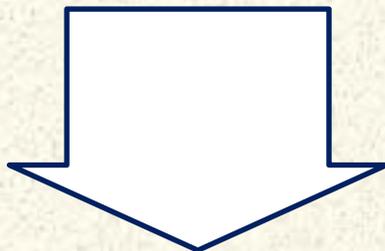
Возможности On-Line при скорости Desktop

Рост быстродействия вычислительных систем

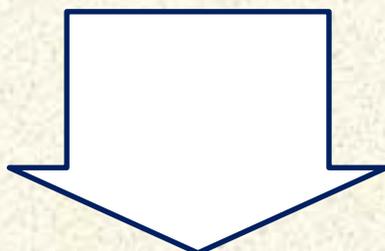


Предпосылки...

Рост размера и функциональности приложений



Связь вероятности ошибки с нетривиальностью кода



Необходимость постоянного обновления приложений

Язык программирования C#

Основы объектно-ориентированного программирования

Основные понятия ООП

1.1 Абстракция. Абстрагирование – это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция – это набор всех таких характеристик.

1.2 Инкапсуляция. Инкапсуляция – это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя.

1.3 Наследование. Наследование – это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. Класс, от которого производится наследование, называется базовым или родительским. Новый класс – потомком, наследником или производным классом.

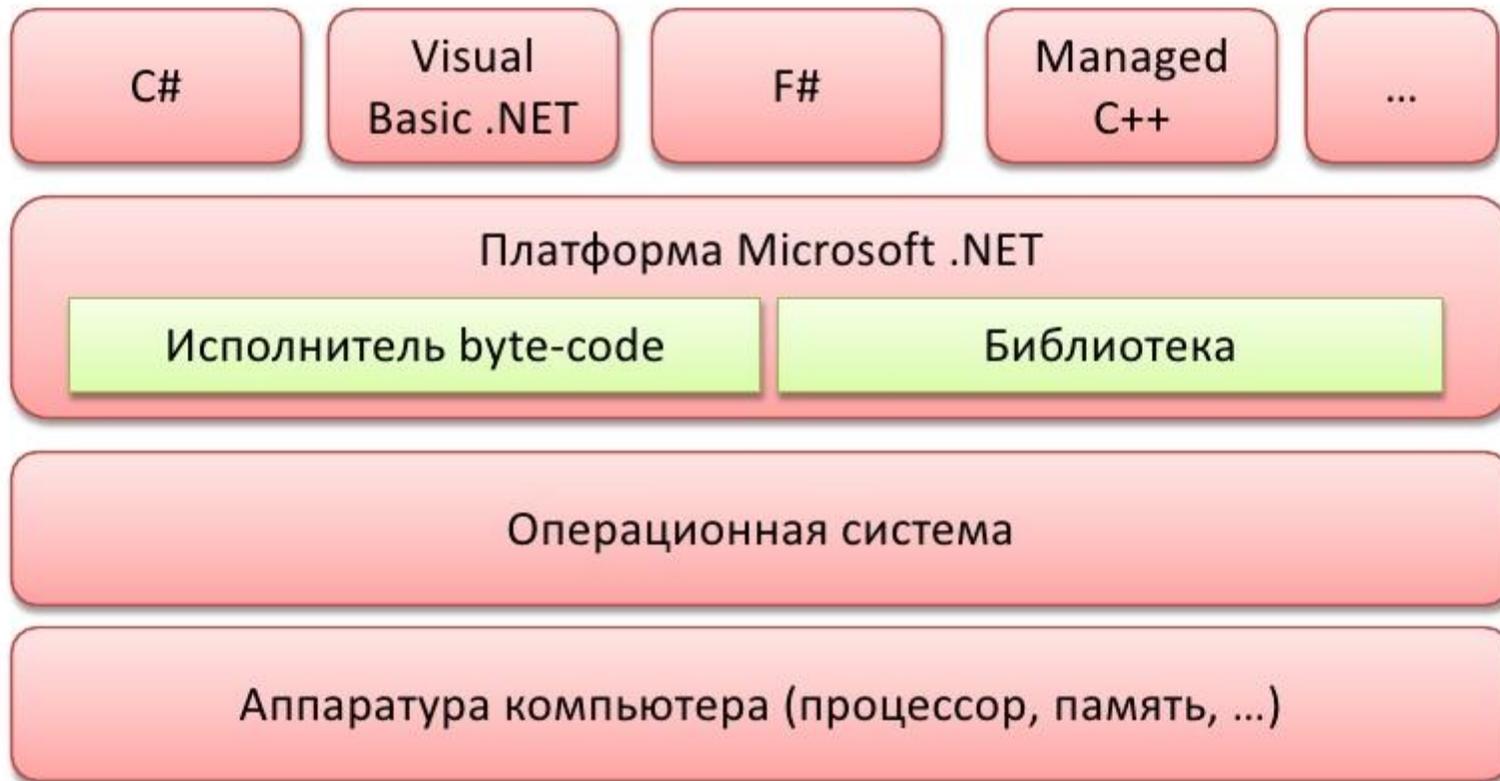
1.4 Полиморфизм. Полиморфизм – это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. Полиморфизм реализуется путём введения в язык правил, согласно которым переменной типа «класс» может быть присвоен объект любого класса-потомка её класса.

1.5 Контроль доступа для методов. Управление областью видимости с помощью модификаторов `public` (для всех), `protected` (для потомков), `private` (для самого класса).

1.6 Методы доступа для контроля диапазонов, возможности записи и чтения - аксессоры (accessor). Реальные поля (атрибуты) - скрытые для исключения неконтролируемого изменения состояния объекта.

1.7 Свойства объекта - “синтаксический сахар”, псевдополя, скрывающие необходимость вызова методов доступа для модификации свойств.

Платформа .NET



Основные понятия dotNET

1.1 Сборка (Assembly) - логическая группировка одного или нескольких управляющих модулей или файлов ресурсов. Может быть исполняемым приложением (.exe) или библиотечным модулем (.dll)

1.2 Управляемый код (Managed Code) - код исполняемый в среде CLR. Код C++ можно сделать управляемым директивой /CLR.

1.3 Управляемые данные (Managed Data) - объекты, размещаемые в управляемой памяти (куче) и уничтожаемые сборщиком мусора CLR. В C# можно использовать и неуправляемые данные.

1.4 Общий кэш сборок (GAC - Global Assembly Cache) - возможность размещения всех необходимых для работы приложения сборок в одном каталоге, возможность регистрации общих сборок (доступных нескольким приложениям).

1.5 Общая система типов (CTS - Common Type System) - структура синтаксических конструкций, способы объявления, использования и управления (преобразования значений) общих типов среды выполнения.

1.6 Пространство имен (Namespace) - способ организации системы типов в единую группу.

Особенности C#

- Мультипарадигмальный язык
 - Императивный
 - Объектно-ориентированный
 - Функциональный
- Что нужно для программирования?
 - .NET Framework SDK
 - Visual Studio Professional/Ultimate (2010)
 - Visual Studio C# Express (2010)

Минимальная программа на C#

```
using System;

namespace MyApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello, World!");
        }
    }
}
```

Переменные и типы данных

- `int` – целый тип
- `float` – вещественный тип
- `double` – вещественный с двойной точностью
- `bool` – логический (значения `true/false`)
- Массивы, ссылки, делегаты, ...
- Объектный
 - `MyObject`, `List<bool>`, ...

Оператор присваивания

```
using System;

namespace MyApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            float a = float.Parse(args[0]);
            float b = float.Parse(args[1]);
            float c = float.Parse(args[2]);
            var d = b * b - 4 * a * c;
            var x1 = -b + Math.Sqrt(d) / 2 / a;
            var x2 = -b - Math.Sqrt(d) / 2 / a;
            Console.WriteLine("x1={0}, x2={1}", x1, x2);
        }
    }
}
```

Операторы передачи управления

- Условный оператор if-then-else
- Циклы
 - С предусловием while ... do
 - С постусловием do ... while
 - Со счётчиком for

Условный оператор

```
using System;

namespace MyApplication
{
    class Program
    {
        static void Main(string[] args)
        {
            float a = float.Parse(args[0]);
            float b = float.Parse(args[1]);
            float c = float.Parse(args[2]);
            var d = b * b - 4 * a * c;
            if (d >= 0)
            {
                var x1 = -b + Math.Sqrt(d) / 2 / a;
                var x2 = -b - Math.Sqrt(d) / 2 / a;
                Console.WriteLine("x1={0}, x2={1}", x1, x2);
            }
            else Console.WriteLine("No solutions");
        }
    }
}
```

Цикл со счётчиком, функции

```
using System;

namespace ConsoleApplication2
{
    class Program
    {
        static int fact(int x)
        {
            int f = 1;
            for (int i = 1; i <= x; i++)
            {
                f = f * i;
            }
            return f;
        }

        static void Main(string[] args)
        {
            for (var i = 1; i < 10; i++)
            {
                Console.WriteLine("{0}!={1}", i, fact(i));
            }
            Console.ReadKey();
        }
    }
}
```

Классы, объекты

```
class Person
{
    public Person(string n) { Name = n; }
    public Person(string n, Person f, Person m)
    {
        Name = n; Father = f; Mother = m;
    }
    public Person Father, Mother;
    public string Name { get; set; }
    public void Introduce()
    {
        Console.WriteLine("I am {0}, my father is {1}, mother is {2}",
            Name, Father.Name, Mother.Name);
    }
}
```

```
var V = new Person("Vasya", new Person("Petya"), new Person("Masha"));
V.Introduce();
```

Полезные классы в библиотеке

- Списки `List<...>`
- Словари `Dictionary<TKey,TValue>`
- Работа с XML, сетевыми подключениями, сжатие данных, криптография, графика, ...

```
var L = new List<Person>();  
  
L.Add(V);  
L.Add(new Person("Jack",V,new Person("Katya")));  
  
L.Sort(new Comparison<Person>((p1, p2) => string.Compare(p1.Name, p2.Name)));  
  
foreach (var x in L) Console.WriteLine(x.Name);  
  
L.ForEach(P => P.Introduce());
```

C# - не только для консольного программирования!

- Windows Forms
- Windows Presentation Foundation

- ASP.NET Web Forms
- ASP.NET MVC

- Silverlight

- XNA Game Studio (Windows, Xbox, Windows Phone 7, Zune)

- Windows Phone 7 (Silverlight, XNA)
- Xbox (XNA)

Университет машиностроения

Кафедра «Автоматика и процессы управления»

Блок дисциплин

Информатика и информационные технологии

Спасибо за внимание !!!

Содержание:

- Технология структурного анализа и проектирования
- UML
- Введение в ООП
- Структура языка C#

...

Контакты:

mami.testolog.ru

timid@mami.ru

inform437@gmail.com