

Программирование в .NET

- Принципы объектно-ориентированного программирования (ООП)
- Классы и объекты в C#

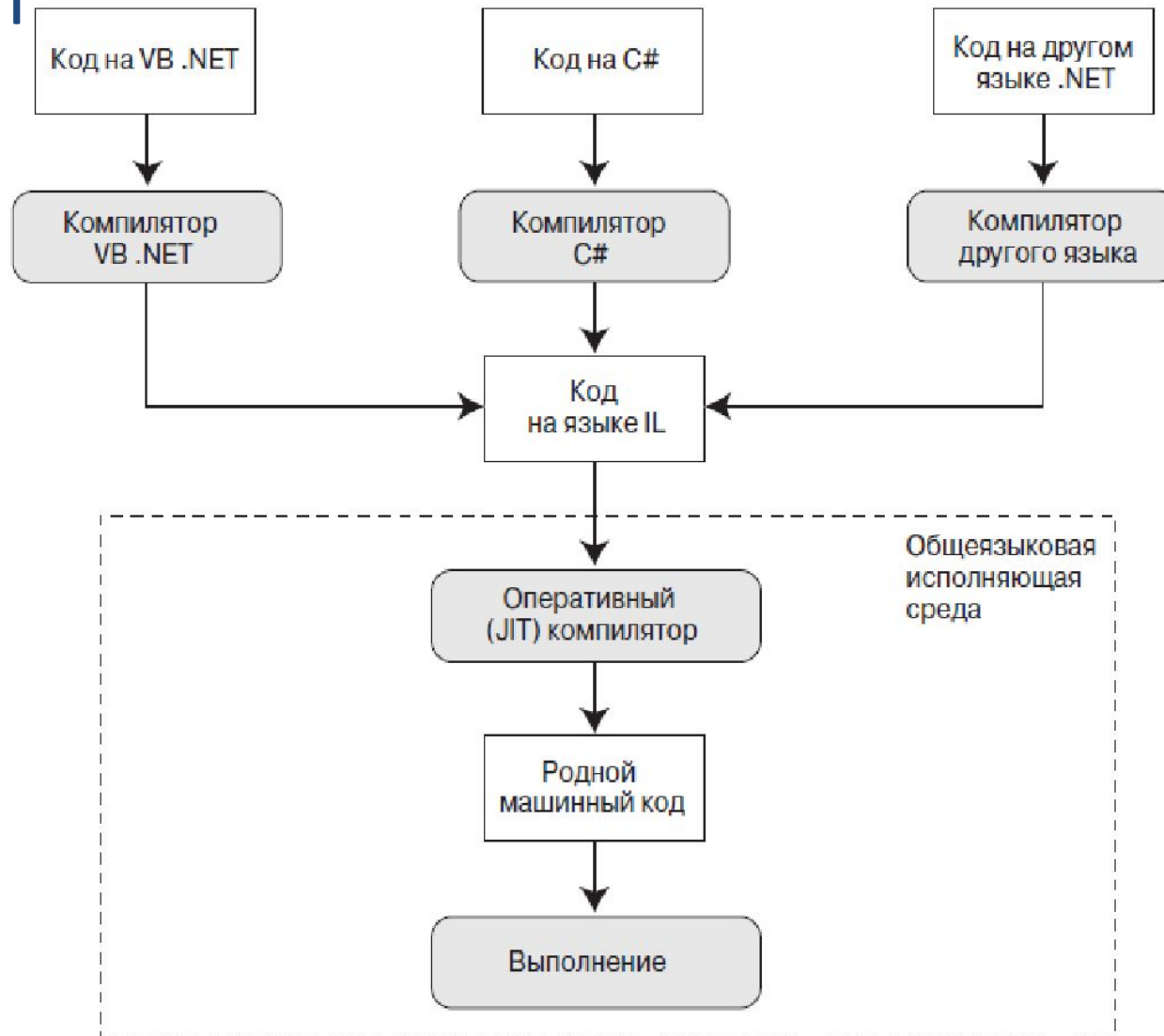
Цель освоения дисциплины

«Программирование в .NET»:

- Научить основам разработки программного и информационного обеспечения с использованием библиотек Framework;
- Научить применению архитектуры .NET для разработки программного обеспечения;
- Способствовать приобретению навыков использования языка C# для программной реализации моделей .NET.

Программирование в

.NET



Объектно-ориентированное программирование

- Объекты
- Классы
- Инкапсуляция
- Наследование
- Полиморфизм

Объектно-ориентированное программирование

Объект – это сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса

Объектно-ориентированное программирование

Процедурное

программирование:

Программирование в

ООП:



класс

ы

Объектно-ориентированное программирование

Объектно-ориентированное, или объектное, программирование (ООП) — парадигма программирования, в которой основными концепциями являются понятия объектов и классов.

C#

Классы

//Пример

класса:
public class SampleClass1
{
}

//Создание нового экземпляра объекта

SampleClass1:
class SampleClass2
{
 SampleClass1 class1 = **new** SampleClass1();
}

C#

Классы

Модификаторы доступа

Public – доступ возможен из любого другого кода в той же сборке или другой сборке, ссылающейся на него

Private – доступ можно получить только из кода в том же классе

Protected – доступ можно получить только из кода в том же классе или в производном классе.

C#

Классы

Поля представляют сведения, содержащиеся в объекте.

Поля подобны переменным в том, что их можно прочитать или изменить напрямую.

```
public class SampleClass1
{
    public string msg;
}
```

C#

Классы

Значение поля может быть получено вне своего класса:

```
class SampleClass2
{
    SampleClass1 class1 = new SampleClass1();
    String msg1 = class1.msg;
}
```

Демонстрационный пример – так не делают!

(нарушение принципов ООП)

C#

Классы

Свойства как и поля представляют сведения, содержащиеся в объекте. Отличие в том, что поля предоставляют прямой доступ к этим сведениям.

```
class SampleClass
{
    public int SampleProperty { get; set; }
}
```

C#

Классы

```
class SampleClass
{
    //объявляем переменную
    private int _sample;
    //создаем свойство для чтения и изменения
    этой переменной
    public int Sample
    {
        get { return _sample; } //на чтение
        set { _sample = value; } //на изменение
    }
}
```

Три кита

ООП

Инкапсуляция или сокрытие реализации является базовым свойством ООП.

Она позволяет создавать пользовательские объекты, обладающие требуемыми методами и далее оперировать ими, не вдаваясь в устройство этих объектов.

Три кита

ООП

Действия объекта называются методами.

```
public class SampleClass
{
    private string msg = "изучаем C#";
    public bool SravnenieStrok(string stroka)
    {
        if (stroka == msg) return true;
        else return false;
    }
}
```

Три кита

ООП

Выполняем метод в другом классе:

```
class SampleClass2
{
    //создаем экземпляр объекта
    SampleClass class2 = new SampleClass();

    //выводим на консоль результат
    Console.WriteLine(
        class2.SravnenieStrok("изучаем Java").ToString() );
}
```


Три кита

ООП

```
//Перемножение двух чисел
public class SampleClass1
{
    private int ch1 = 5;

    public int Multiplay (int ch2)
    {
        return ch1 * ch2;
    }
}
```

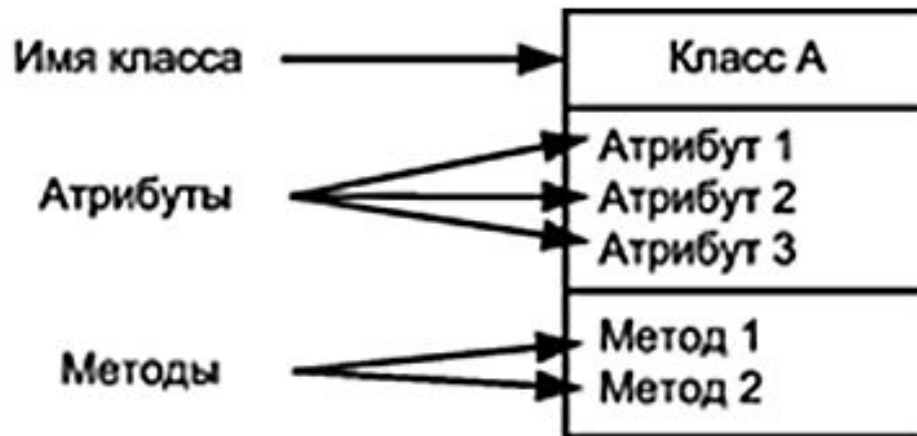
Три кита

ООП

```
//получаем доступ к методу другого класса
class SampleClass2
{
    SampleClass1 class2 = new SampleClass1();
    Console.WriteLine(class2. Multiplay(10).ToString());
}
```

Три кита ООП

При **наследовании** все атрибуты и методы родительского класса наследуются классом-потомком.



а)

Три кита

ООП

Наследование — это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствуемой функциональностью.

Класс, от которого производится наследование, называется базовым (родительским или суперклассом).

Новый класс — потомком (наследником

Три кита

ООП

```
// класс SampleClass2 является наследником  
// класса SampleClass1
```

```
class SampleClass2 : SampleClass1  
{  
  
}
```

Три кита

ООП

Полиморфизм – это выполнение методом с одним и тем же именем различных действий в зависимости от контекста, в частности, от принадлежности тому или другому классу.

```
//перебираем все элементы объекты  
(формы)  
foreach (Control cntrl in this.Controls)  
{  
    //окрашиваем шрифт в красный цвет  
    cntr.ForeColor = Color.Red;  
}
```

C#

Циклы

```
switch (comboBox1.Text)
{
    case "Красный":
        this.BackColor = Color.Red;
        break;
    case "Желтый":
        this.BackColor = Color.Yellow;
        break;
    case "Зеленый":
        this.BackColor = Color.Green;
        break;
    default:
        this.BackColor = Color.Gray;
        break;
}
```

Лабораторная

работа 1

1. Составить программу решения линейного уравнения $ax + b = 0$, где a не равно 0.
2. Определить максимальное из двух различных вещественных чисел.
3. Для натурального числа k напечатать фразу: «Мы нашли k грибов в лесу», согласовав окончание слова «гриб» с числом k .
4. Даны площади круга и квадрата. Определить: 1) уместится ли квадрат в круге? 2) уместится ли круг в квадрате?
5. Даны две скорости: одна в километрах в час, другая – в метрах в секунду. Какая из скоростей больше?
6. Известны год и номер месяца рождения человека, а также год и номер месяца сегодняшнего дня. Определить возраст человека (число полных лет). В случае совпадения номеров месяцев считать, что прошел полный год.
7. Дана фамилия человека. Вывести приветствие. В случае, когда фамилия оканчивается на «ов» - «Здравствуйте господин Иванов!», на «ова» - «Здравствуйте госпожа Иванова!» В остальных случаях вывести: «Здравствуйте господин(госпожа) ...».
8. Даны дни недели. Вывести в зависимости от выбора, какие занятия на этот день запланированы.