

Курс «С#. Программирование на языке высокого уровня»

Павловская Т.А.

Лекция 1. Начальные сведения

Первое представление об основных понятиях объектно-ориентированного программирования, платформе .NET и входящей в ее состав среде разработки Visual Studio .NET.

Литература

1. Стандарт языка:
<http://www.ecma-international.org/publications/standards/Ecma-334.htm>
2. Павловская Т.А. С#. Программирование на языке высокого уровня. Учебник — СПб.: ПИТЕР, 2010. — 432 с.
3. Троелсен Э. С# и платформа .NET. Библиотека программиста. — СПб.: Питер, 2008. — 796 с.
4. Нэш Т. С# 2008: ускоренный курс для профессионалов. — М.: «Вильямс», 2008. — 576 с.
5. Нейгел К., Ивьен Б., Глинн Д., Уотсон К., Скиннер М. С# 2008 и платформа .NET 3.5 для профессионалов (или: С# 2005 и платформа .NET 3.0 для профессионалов) — М.: «Вильямс», 2008. — 1376+416 (на CD) с.
6. Любые другие книги с символами "С#" на обложке!

Скачать среду С#: <http://msdn.microsoft.com/vstudio/express/visualCsharp/>

Среда разработки и платформа .NET

Среда разработки обычно содержит:

- текстовый *редактор*;
- *компилятор*;
- средства *отладки и запуска программ*;
- *общие библиотеки*;
- *справочную систему*
- и другие элементы.

Платформа .NET

- Среда разработки Visual Studio.NET для нескольких языков программирования
- Каркас .NET Framework
 - библиотека классов
 - исполняющая система (виртуальная машина)
- Механизмы поддержки баз данных, электронной почты и коммерции и множество других средств

Структура .NET

Платформа .NET

Среда разработки приложений **Visual Studio.NET**

Каркас **.NET Framework**

Библиотека классов каркаса (FCL)

Web-сервисы

Интерфейс пользователя

Классы для работы с данными и XML

Базовые классы среды

Общезыковая среда выполнения (CLR)

**.NET Enterprise
Servers**

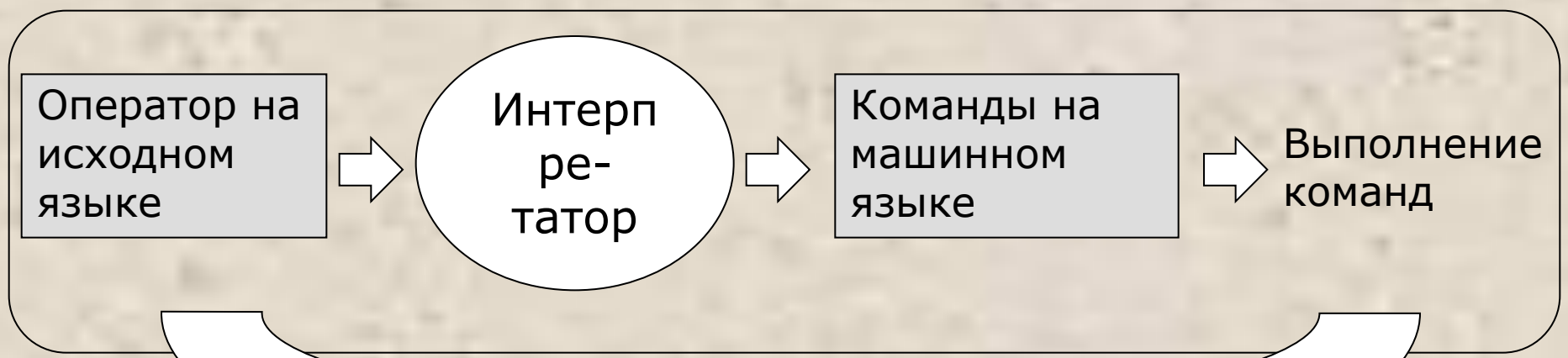
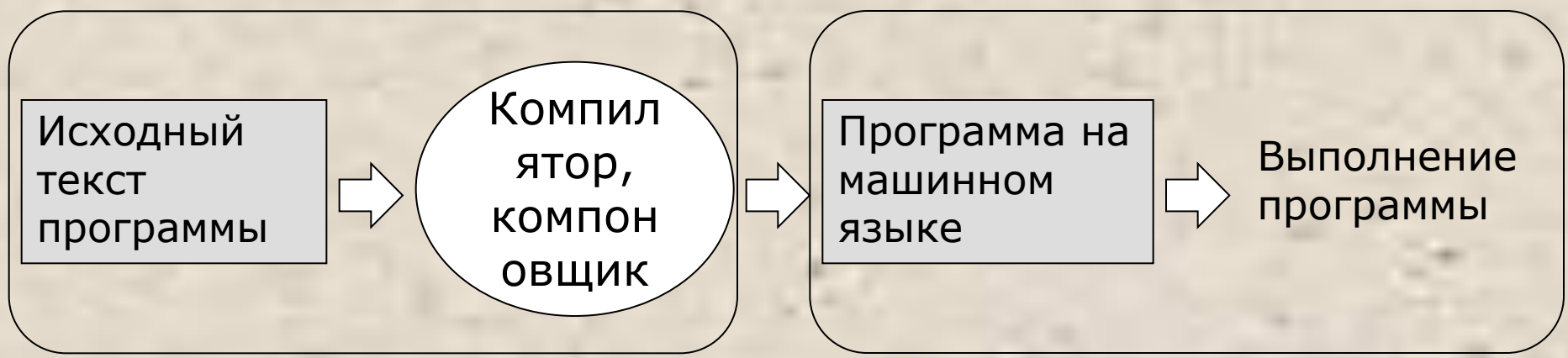
**.NET Building
Block Services**

Операционная система

Трансляция

Компиляция

Интерпретация



Гибридная схема трансляции

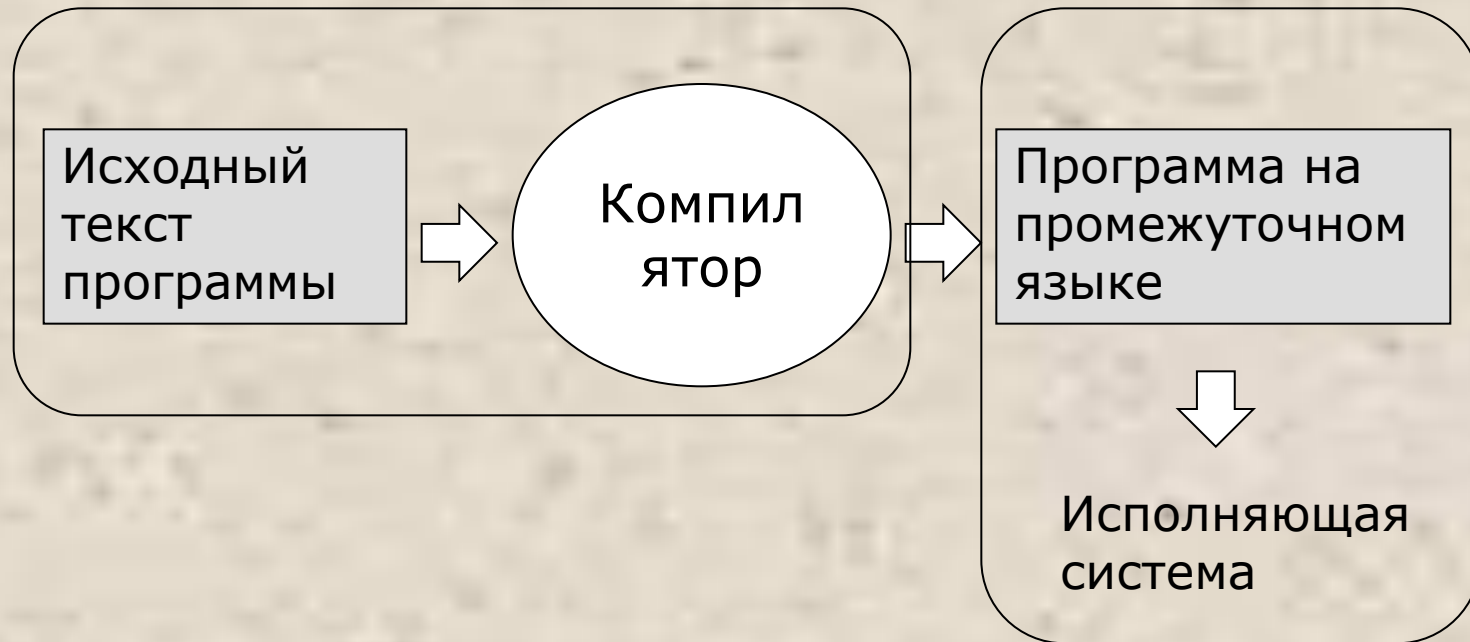
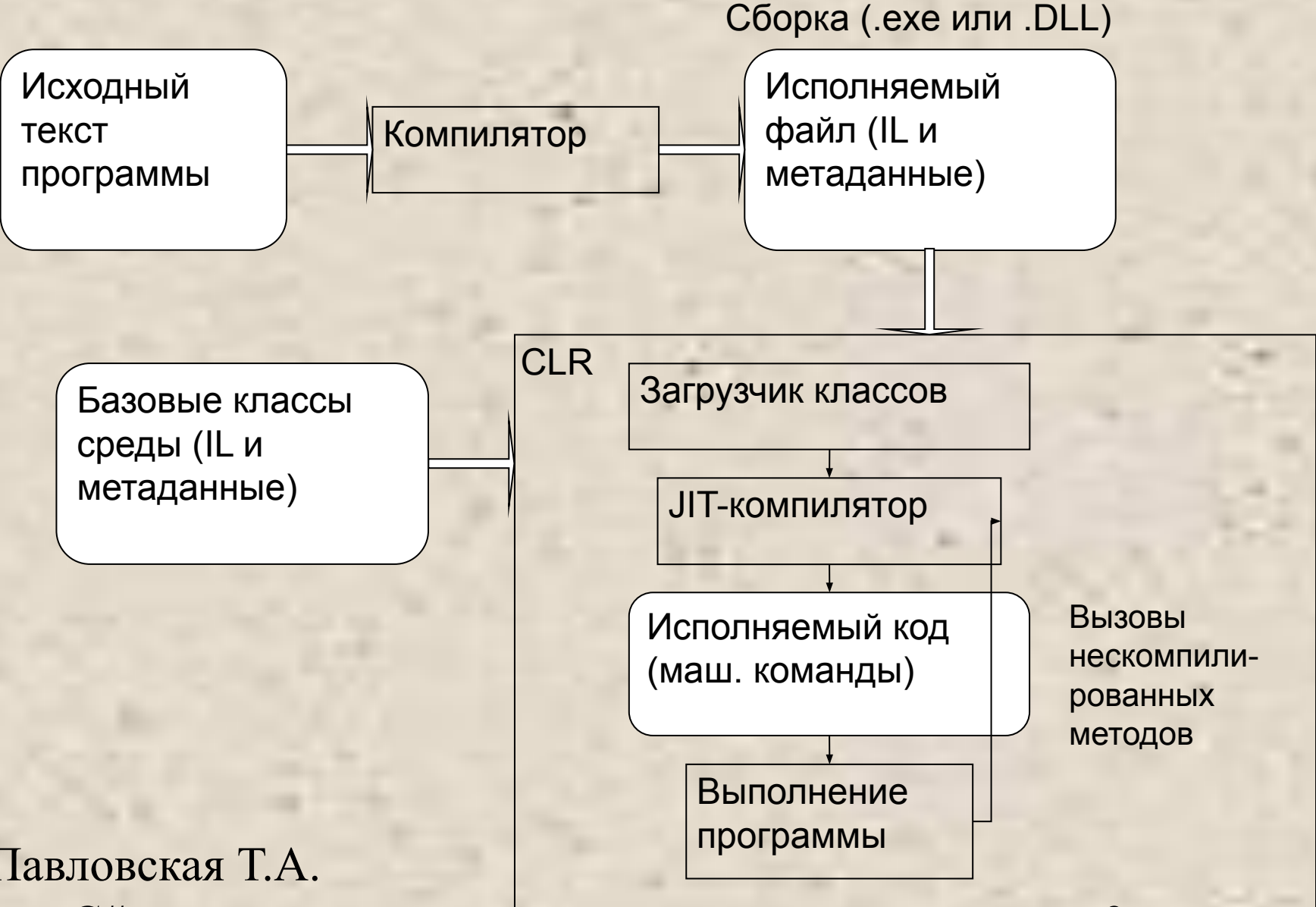


Схема выполнения программы в .NET



Основные критерии качества программы

- надежность
- возможность точно планировать производство и сопровождение

Для достижения этих целей программа должна:

- иметь простую структуру
- быть хорошо читаемой
- быть легко модифицируемой

Парадигмы программирования

Парадигма — способ организации программы, принцип ее построения. Наиболее распространенными являются процедурная и объектно-ориентированная парадигмы.

Они различаются способом декомпозиции, положенным в основу при создании программы.

Процедурная декомпозиция состоит в том, что задача, реализуемая программой, делится на подзадачи, а они, в свою очередь — на более мелкие этапы, то есть выполняется пошаговая детализация алгоритма решения задачи.

Объектно-ориентированная декомпозиция предполагает разбиение предметной области на объекты и реализацию этих объектов и их взаимосвязей в виде программы.

Первый взгляд на классы

- Понятие *класс* аналогично обыденному смыслу этого слова в контексте «класс членистоногих», «класс задач».
- Класс является обобщенным понятием, определяющим характеристики и поведение некоторого множества конкретных объектов этого класса, называемых *экземплярами класса (объектами)*.
- Класс содержит *данные*, задающие свойства объектов класса, и *функции (методы)*, определяющие их поведение.
- Все классы .NET имеют одного общего предка — класс *object*, и организованы в единую иерархическую структуру.
- Классы логически сгруппированы в так называемые *пространства имен*, которые служат для упорядочивания имен классов и предотвращения их конфликтов: в разных пространствах имена могут совпадать. Пространства имен могут быть вложенными

©Павловская Т.А.

Заготовка консольной программы

```
using System;
namespace ConsoleApplication1
{
class Class1
    {
static void Main(string[] args)
    {
        //
        // TODO: Add code to start application here
        //
    }
}
}
```