

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

КАФЕДРА АВТОМАТИЗАЦИИ ПРЕДПРИЯТИЙ СВЯЗИ

Web-технологии в управлении техническими системами

2019

РАЗДЕЛ 1. ВВЕДЕНИЕ В ВЕБ-ТЕХНОЛОГИИ УПРАВЛЕНИЯ ТЕХНИЧЕСКИМИ СИСТЕМАМИ

Инструментарий

Visual Studio 2019 Community

SQL Server 2017 Express

Microsoft SQL Server Management Studio 2017

Эрик Эванс

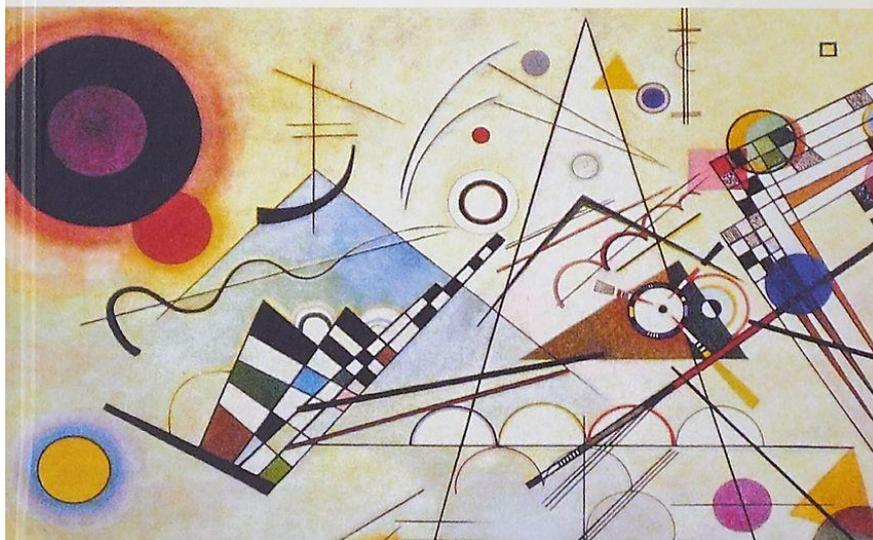
Предисловие
Мартина Фаулера

Domain-Driven

DESIGN

Предметно-ориентированное проектирование

СТРУКТУРИЗАЦИЯ СЛОЖНЫХ
ПРОГРАММНЫХ СИСТЕМ



Джимми Нильссон



Применение DDD и шаблонов проектирования

Проблемно-ориентированное
проектирование приложений
с примерами на C# и .NET

Предисловия Мартина Фаулера и Эрика Эванса

The Addison-Wesley Signature Series



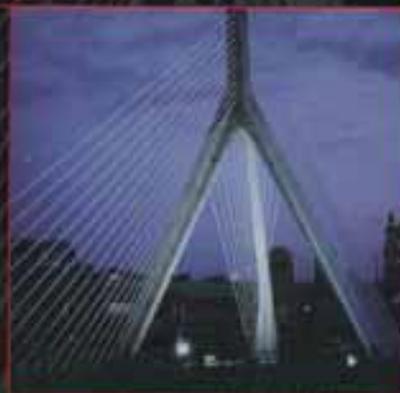
АРХИТЕКТУРА КОРПОРАТИВНЫХ ПРОГРАММНЫХ ПРИЛОЖЕНИЙ



ИСПРАВЛЕННОЕ ИЗДАНИЕ

МАРТИН ФАУЛЕР

ПРИ УЧАСТИИ
ДЕЙВИДА РАЙСА,
МАТТЬЮ ФОММЕЛА,
ЭДВАРДА ХАЙЕТА,
РОБЕРТА МИ
И РЭНДИ СТАФФОРДА





ОБЪЕКТНО- ОРИЕНТИРОВАННЫЙ АНАЛИЗ И ПРОЕКТИРОВАНИЕ С ПРИМЕРАМИ ПРИЛОЖЕНИЙ

Третье издание

ГРАДИ БУЧ, РОБЕРТ А. МАКСИМЧУК,
МАЙКЛ У. ЭНГЛ, БОББИ ДЖ. ЯНГ,
ДЖИМ КОНАЛЛЕН,
КЕЛЛИ А. ХЬЮСТОН

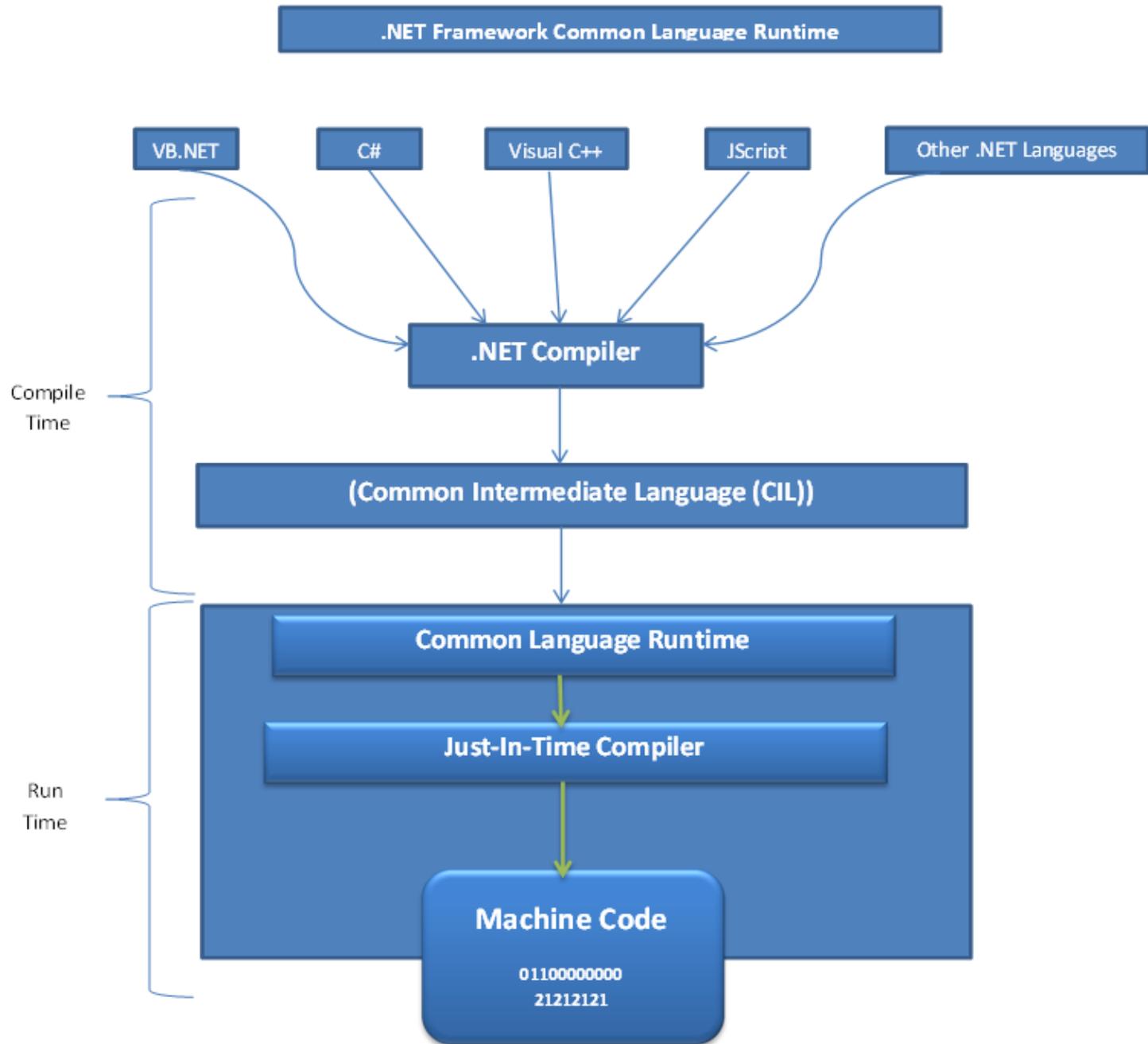


Интернет и всемирная паутина

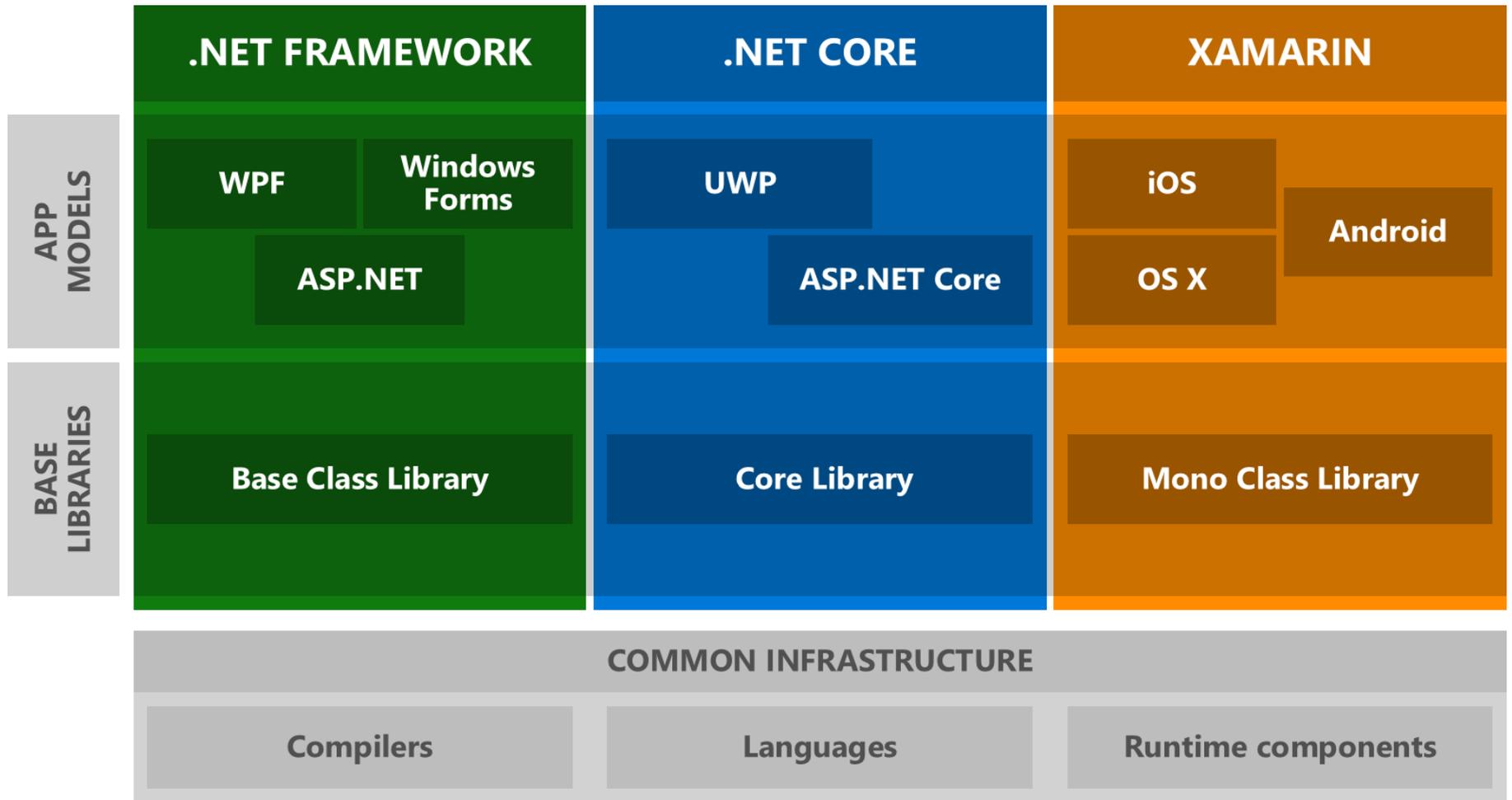
Интернет и Всемирная паутина

Интернет (англ. Internet) – всемирная система объединённых компьютерных сетей для хранения, обработки и передачи информации.

Всемирная паутина (англ. World Wide Web) – распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключённых к сети Интернет. Для обозначения Всемирной паутины также используют слово веб (англ. web «паутина») и аббревиатуру WWW.



Введение в ASP.NET Core



.NET FRAMEWORK

.NET CORE

XAMARIN

APP MODELS

WPF

Windows Forms

UWP

ASP.NET Core

iOS

Android

OS X

ASP.NET

BASE LIBRARIES

Base Class Library

Core Library

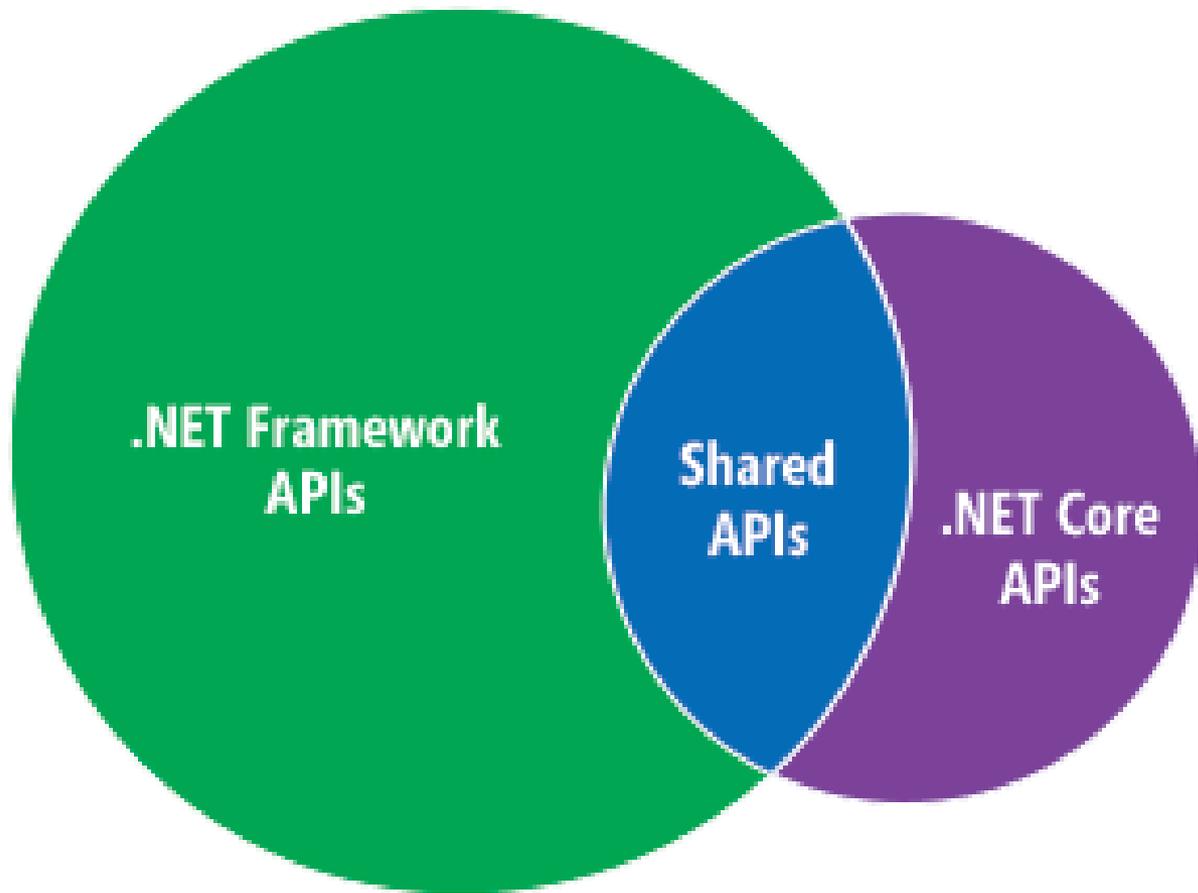
Mono Class Library

COMMON INFRASTRUCTURE

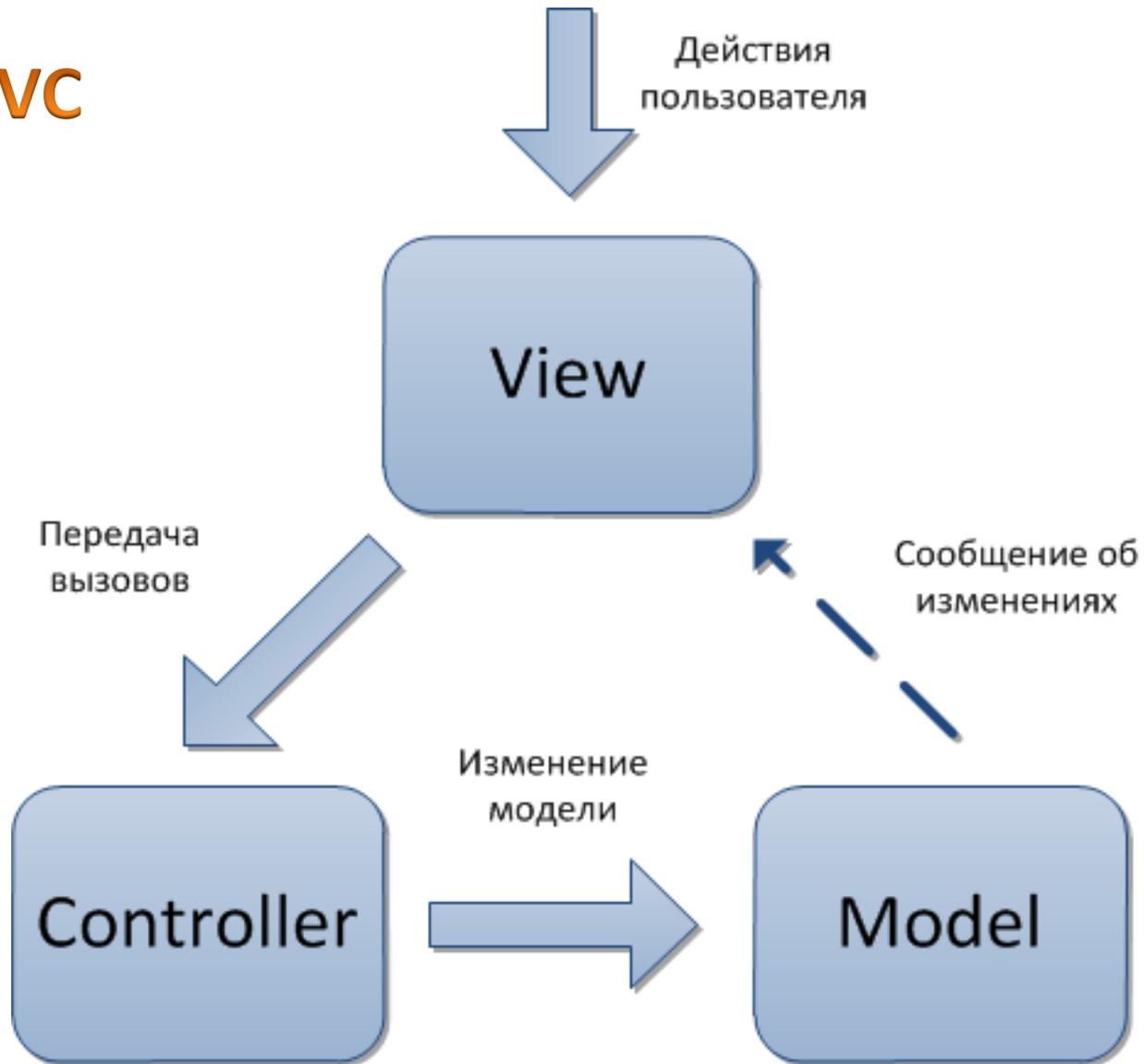
Compilers

Languages

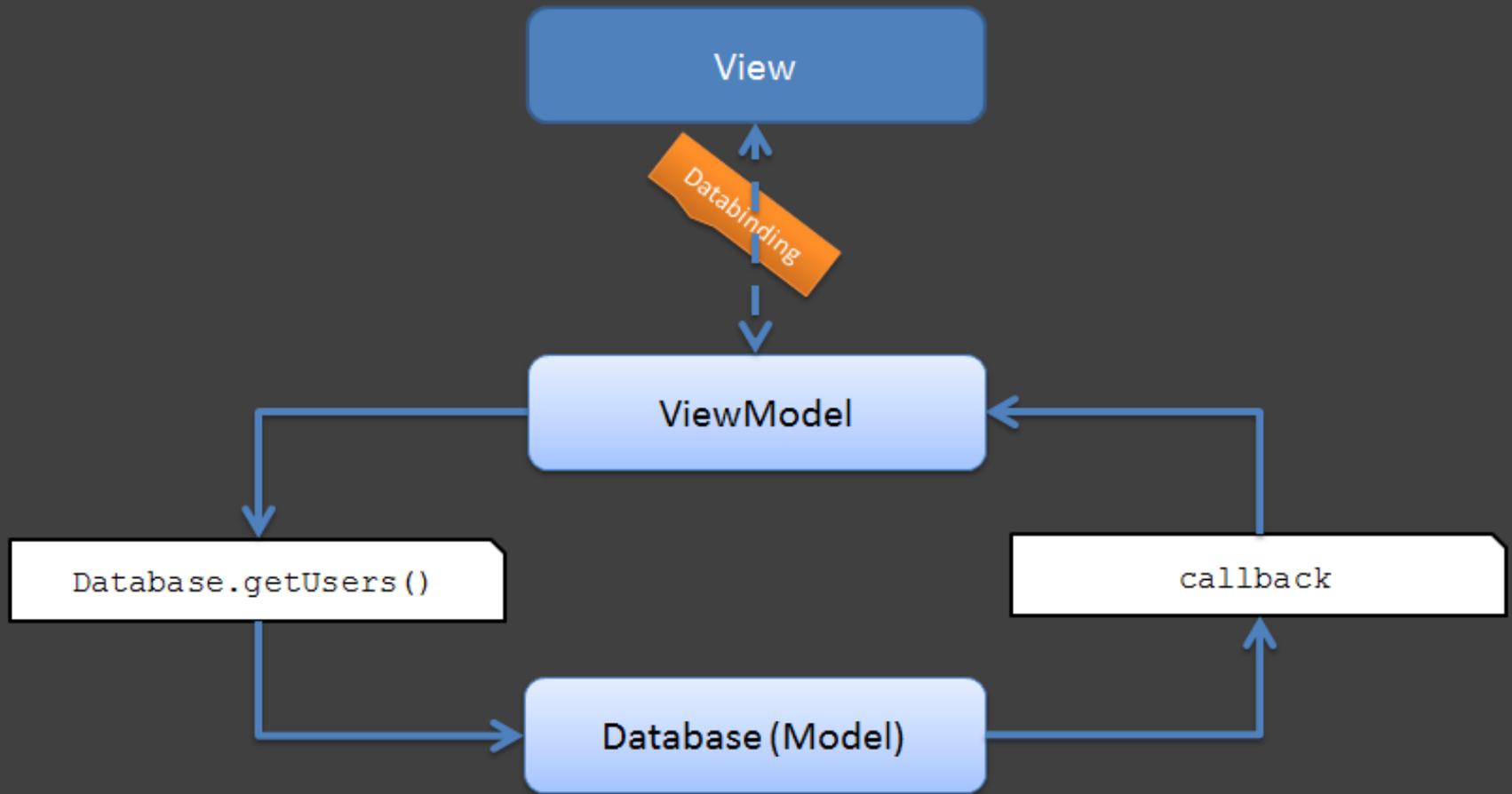
Runtime components



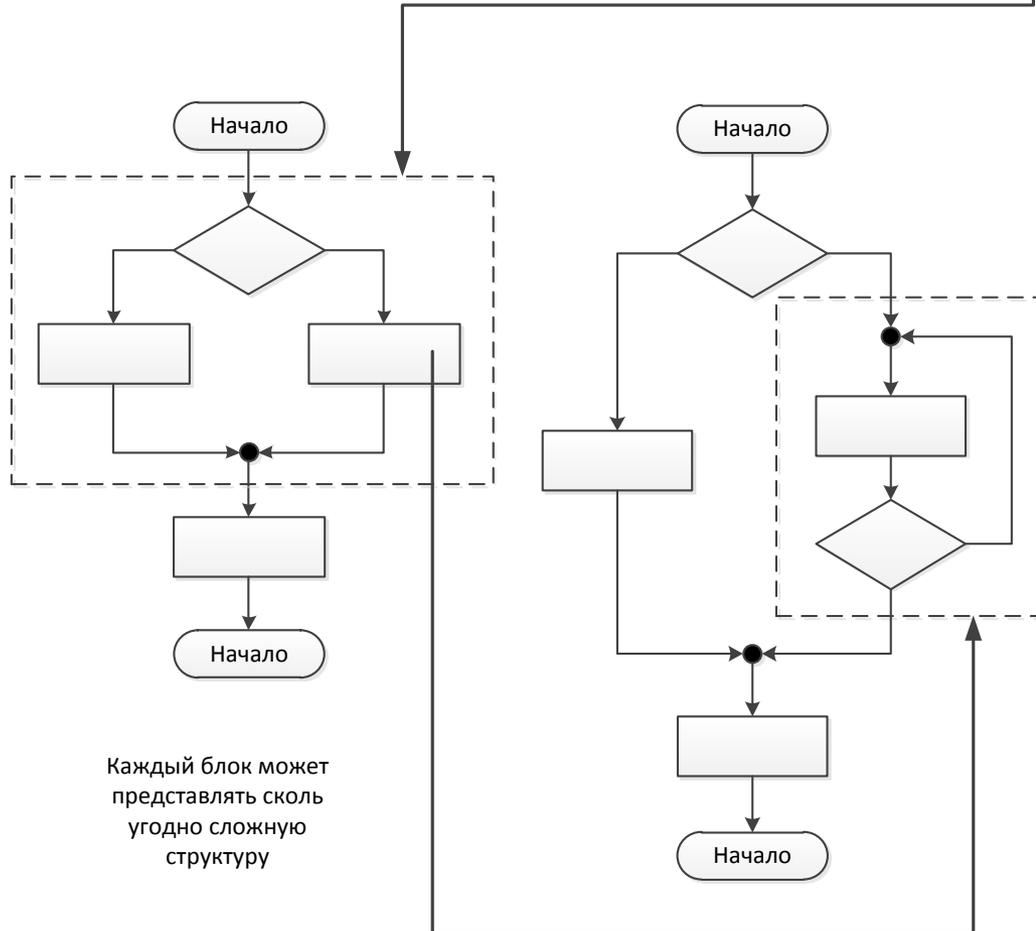
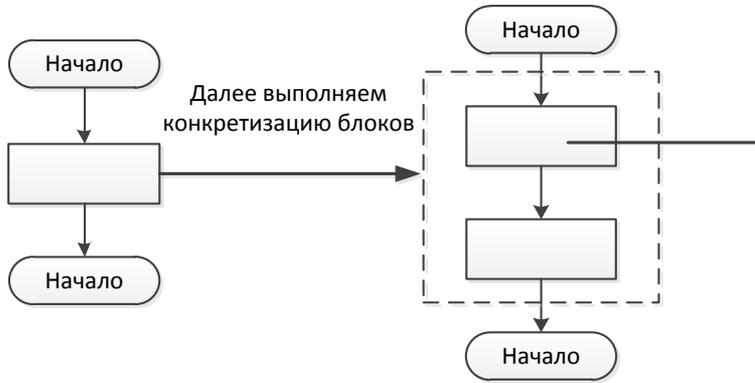
MVC



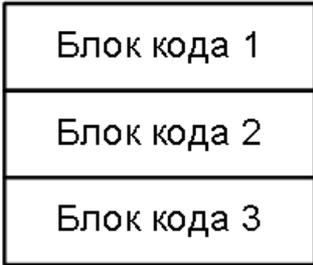
MVVM



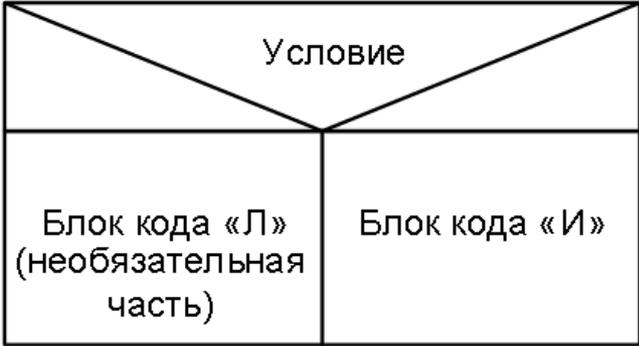
Первоначально всю программу можно представить в виде одного блока



Диаграммы Насси-Шнайдермана



Последовательное выполнение алгоритма



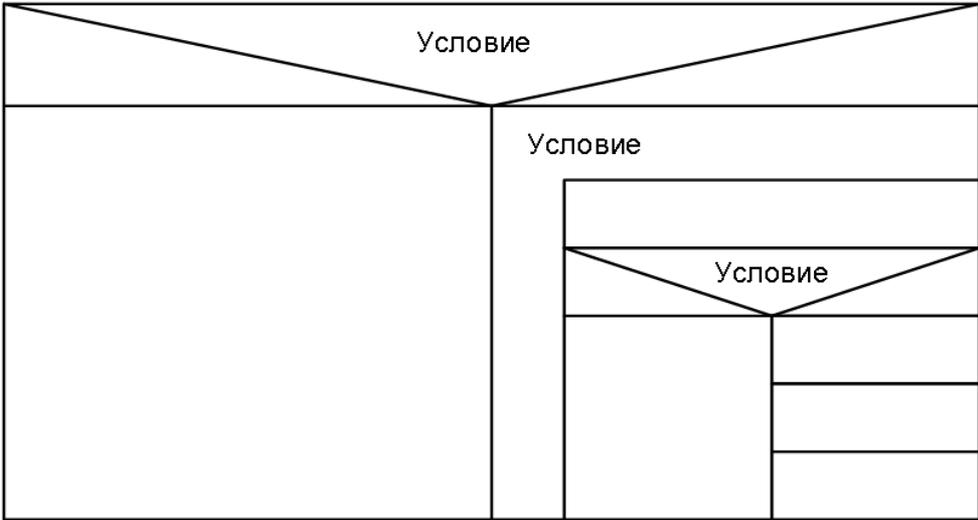
Ветвящаяся структура



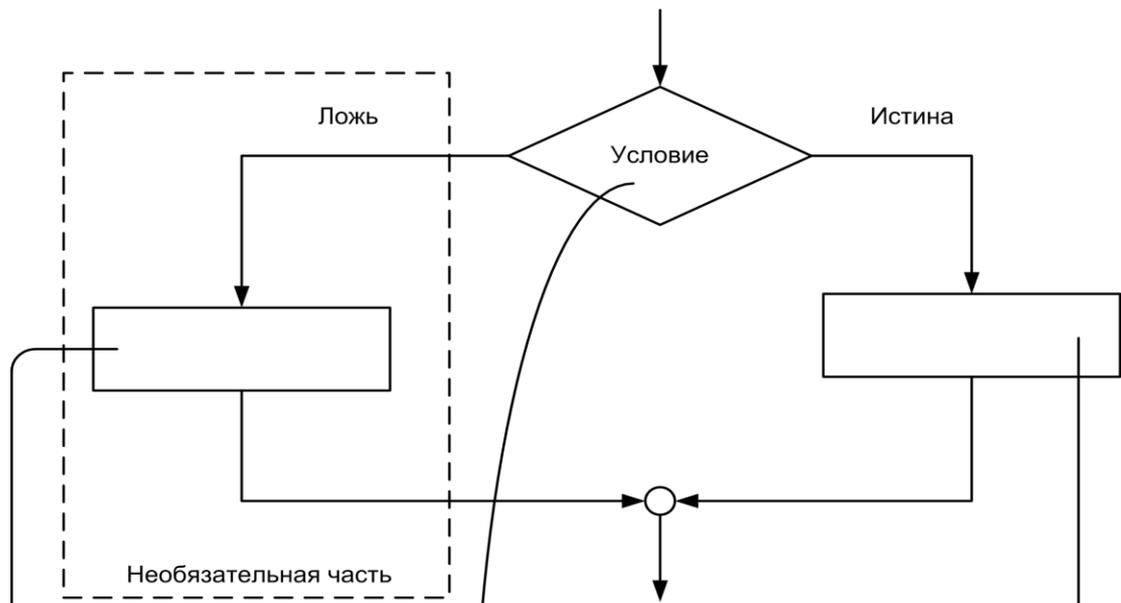
Цикл с предусловием



Цикл с постусловием



Пример вложенных структур диаграмм Насси-Шнайдермана



a)

```

if(      )
{

```

.....


```

}
```

```

else
{

```

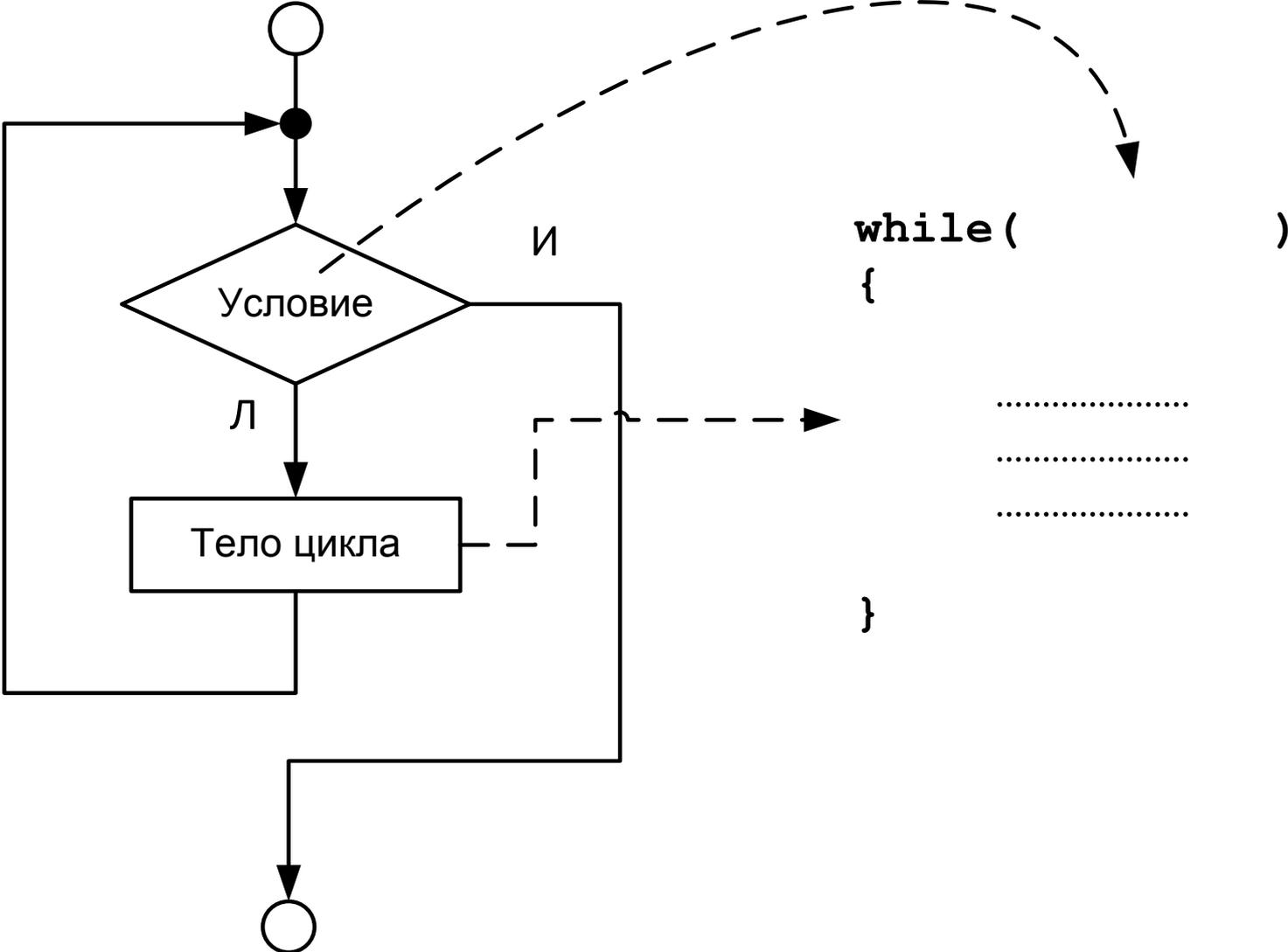
.....

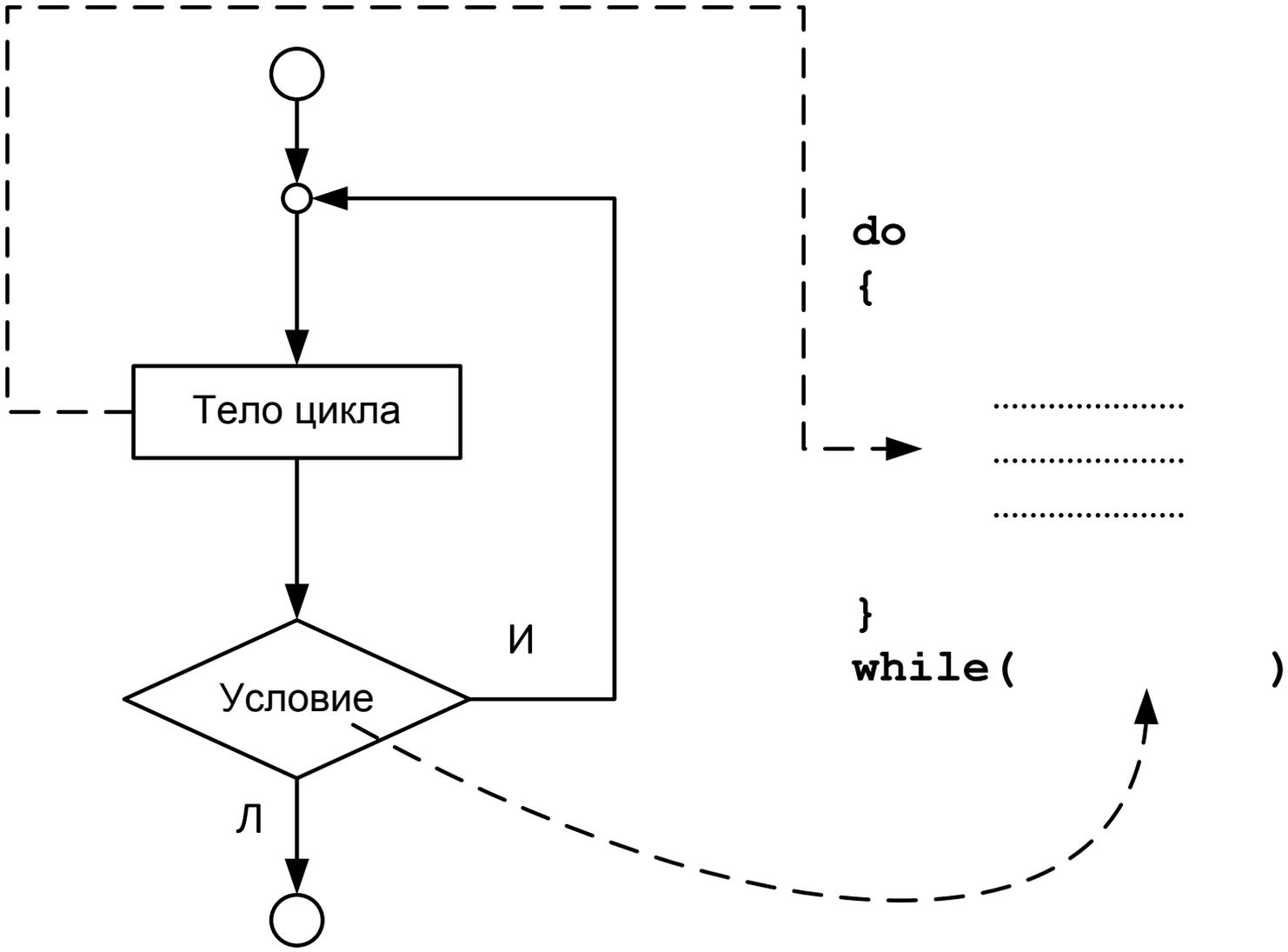

```

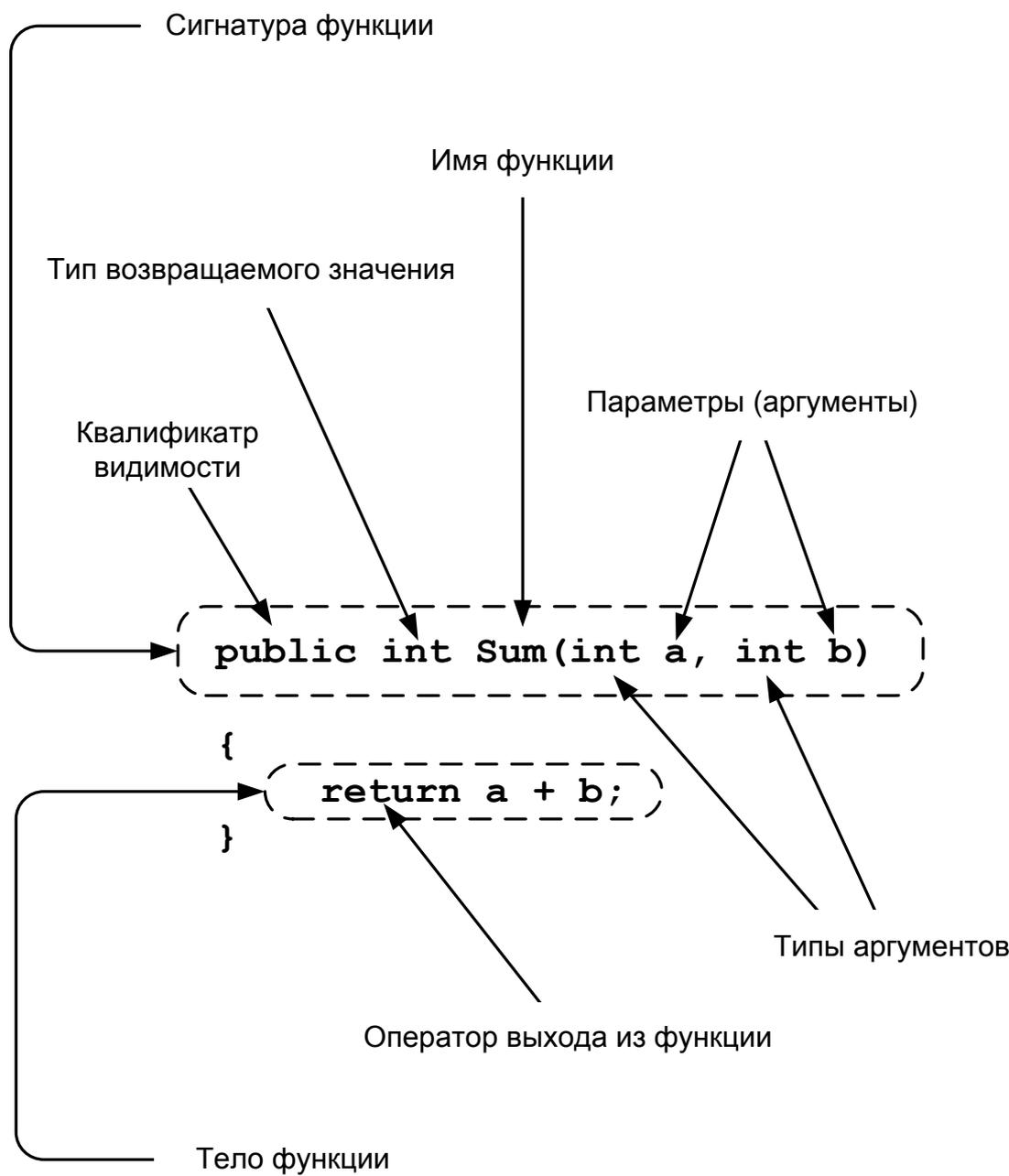
}
```

Необязательная часть

б)







Классы и объекты

Конструктор с параметрами

```
class Point
{
    ссылка: 8
    public double X { get; set; }
    ссылка: 8
    public double Y { get; set; }

    ссылка: 2
    public Point (double x, double y)
    {
        X = x;
        Y = y;
    }

    ссылка: 1
    public void MoveBy(double dx, double dy)
    {
        X += dx;
        Y +=dy;
    }
}
```

Перегрузка конструктора

```
class Point
{
    ссылка: 9
    public double X { get; set; }
    ссылка: 9
    public double Y { get; set; }

    ссылка: 2
    public Point()
    {
        X = 0;
        Y = 0;
    }

    ссылка: 0
    public Point (double x, double y)
    {
        X = x;
        Y = y;
    }

    ссылка: 1
    public void MoveBy(double dx, double dy)
    {
        X += dx;
        Y += dy;
    }
}
```

Создание объектов с помощью конструктора с параметрами и без параметров

```
static void Main(string[] args)
{
    Point p1 = new Point();
    p1.X = 5;
    p1.Y = 10;
    Console.WriteLine("P1.X = {0}, P1.Y = {1}", p1.X, p1.Y);

    Point p2 = new Point(15, 20);
    Console.WriteLine("P2.X = {0}, P2.Y = {1}", p2.X, p2.Y);

    p1.MoveBy(50, 100);
    Console.WriteLine("P1.X = {0}, P1.Y = {1}", p1.X, p1.Y);
    Console.WriteLine("P2.X = {0}, P2.Y = {1}", p2.X, p2.Y);
}
```

```
class Program
{
    ссылка: 0
    static void Main(string[] args)
    {
        Point p1 = new Point(5, 10);
        Point p2 = new Point(p1.X, p1.Y);
        Console.WriteLine(p1.Position);
    }
}
```

Копирующий конструктор

```
public Point (Point point)
{
    X = point.X;
    Y = point.Y;
}
```

```
static void Main(string[] args)
{
    Point p1 = new Point(5, 10);
    Point p2 = new Point(p1.X, p1.Y);
    Point p3 = new Point(p2);

    Console.WriteLine(p3.Position);
}
```

Метод клонирования объекта

```
public Point Clone()  
{  
    return new Point(X, Y);  
}
```

```
X = 5, Y = 10  
X = 5, Y = 10  
X = 5, Y = 10  
X = 5, Y = 10
```

```
static void Main(string[] args)  
{  
    Point p1 = new Point(5, 10);  
    Point p2 = new Point(p1.X, p1.Y);  
    Point p3 = new Point(p2);  
    Point p4 = p3.Clone();  
  
    Console.WriteLine(p1.Position);  
    Console.WriteLine(p2.Position);  
    Console.WriteLine(p3.Position);  
    Console.WriteLine(p4.Position);  
}
```

