



Программирование на языке высокого уровня

Буяновская Елизавета
Михайловна

Lee.buyanovskaya@gmail.com

921-971-2302

Лекция 1

- Содержание курса
- Классификация языков программирования
- Парадигма объектно-ориентированного программирования (ООП).
- Язык программирования JAVA (история развития и особенности)
- Первая программа на JAVA (запуск из командной строки)
- Знакомство со средами разработки
- Введение в классы/объекты, примеры

Содержание курса

- 15 недель: 2 часа лекций/3 часа лабораторных работ
- 4 лабораторных работы (рабочий код, защита; отчет содержит исходный код с комментариями, структуру классов, если необходимо, результат работы программы)
- 4 теста – допуск к л/р
- экзамен

Содержание курса

- Введение, объектно-ориентированное программирование
- Лексика языка, типы данных, пакеты
- Объявление классов, модификаторы доступа
- Особенности объектной модели Java
- Массивы, условные операторы, циклы
- Работа с исключениями
- Многопоточность

Уровни языков программирования

- «Железное» программирование
Поведение программируемого устройства задается путем операций с физическими связями его компонентов.
- Машинные языки программирования
Программа представляет собой последовательность элементарных инструкций для центрального процессора.
- Машинно-ориентированные языки (ассемблеры)
Язык содержит мнемонические команды, операторы для управления ходом исполнения, макрокоманды.
- Языки программирования высокого уровня
Язык в определенном смысле приближен к естественному человеческому языку, обладает сложной грамматикой, средствами управления ходом исполнения и т.п.

Языки программирования

- Функциональные (LISP)
- Процедурные (императивные) (C, Pascal, Fortran)
- Визуальные (UML, BPEL)
- Декларативные (SQL)
- Описания интерфейсов (IDL, WSDL)
- Объектно-ориентированные (C++, C#, Java)
- Логические (ProLog)
- Сценарные (скриптовые) (Forth, Perl, PHP)
- Эзотерические

Языки программирования

- **Функциональные (LISP)**

Алгоритм реализуется как последовательность вычисления значений функций. Программа является stateless – состояние в явном виде не хранится.

- **Процедурные (императивные) (C, Pascal, Fortran)**

Программа представляет собой совокупность конечных автоматов, процесс ее исполнения состоит в изменении этих состояний.

- **Визуальные (UML, VPEL)**

Языки предназначены для графического представления процессов, структур, взаимосвязей.

- **Декларативные (SQL)**

Разработчик описывает ожидаемый результат исполнения программы, не налагая ограничений на способ и процесс получения этого результата средой программирования.

Языки программирования

- **Описания интерфейсов (IDL, WSDL)**
Языки описывают интерфейсы и способы взаимодействия компонентов сложных комплексов, предназначены для «гладкой» интеграции разнородных элементов архитектуры.
- **Объектно-ориентированные (C++, C#, Java)**
Языки основаны на объектно-ориентированной парадигме программирования: вычислительная задача рассматривается как совокупность объектов (с их свойствами и состояниями) и отношений между ними.
- **Логические (ProLog)**
Программа представляет собой совокупность логических связей, процесс исполнения – синтез более сложных логических отношений на основе заданных «простых».

Языки программирования

- Сценарные (скриптовые) (Forth, Perl, PHP)

Языки оперируют понятиями и примитивами некоторой заранее созданной системы, ориентированы на создание сценариев ее взаимодействия с пользователем.

- Эзотерические

Языки, не предназначенные для практического применения. Чаще всего используются для яркого отражения какой-либо идеи или концепции программирования.

Intercal – максимальное отличие от «нормальных» языков

Brainfuck – максимальное сокращение синтаксиса

Befunge – многомерное представление программ, цветовое кодирование и т.п.

Malbolge – максимальное осложнение разработки и чтения кода

Shakespeare, Chef – литературный синтаксис

Объектно-ориентированное программирование

- Программа представляет собой совокупность объектов, для которых определены свойства, функции и поведение. Есть только объекты!
- Исполнение программы – обмен сообщениями между объектами. Сообщение – требование на исполнение действия и аргументы.
- Каждый объект обладает независимой от других объектов памятью.

Понятия ООП

- **Класс** — тип, описывающий устройство объектов: структуру, поведение и способ представления. Класс - чертеж или шаблон, по которому создаются объекты. Обычно классы разрабатывают таким образом, чтобы их объекты соответствовали объектам предметной области.
- **Объект** — сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса (например, после запуска программы на выполнение).

Объектно-ориентированное программирование

- Каждый объект является экземпляром класса, который выражает общие свойства объектов.
- В классе задаётся поведение (функциональность) объекта. Все объекты, которые являются экземплярами одного класса, могут выполнять одни и те же действия.
- Классы организованы в единую древовидную структуру с общим корнем, называемую иерархией наследования.

«Три кита» ООП

- **Абстракция данных**

Объекты представляют собою упрощенное, идеализированное описание реальных сущностей предметной области. Если соответствующие модели **адекватны** решаемой задаче, то работать с ними оказывается намного удобнее, чем с низкоуровневым описанием всех возможных свойств и реакций объекта.

- **Инкапсуляция**

Любой класс рассматривается как чёрный ящик — пользователь класса видит и использует только интерфейсную часть класса (т. е. список декларируемых свойств и методов класса), не вникая во внутреннюю реализацию. Принцип инкапсуляции **теоретически** позволяет минимизировать число связей между классами и упростить независимую реализацию классов.

«Три кита» ООП

- **Наследование**

Наследованием называется возможность порождать один класс от другого с сохранением всех свойств и методов класса-предка, добавляя новые свойства и методы. Набор классов, связанных отношением наследования, называют иерархией.

- **Полиморфизм**

Функции (методу) с одним и тем же именем соответствует разный программный код (полиморфный код) в зависимости от того, объект какого класса используется при вызове данного метода. Такой механизм называется **динамическим (или поздним) связыванием** — в отличие от статического (раннего) связывания, осуществляемого на этапе компиляции.

Особенности и преимущества ООП

- **Преимущества**

- Наглядность описания
- Удобство работы с кодом
- Ориентированность на создание сложных многокомпонентных Enterprise-систем
- Удобство совместной разработки
- Актуальность
- Безопасность для неопытного разработчика.

Особенности и преимущества ООП

- **Особенности**

- Проблема хрупкости базового класса
- Немотивированное усложнение иерархии
- Необходимость внимательного обращения с полиморфными структурами
- Требовательность к объему памяти системы
- Необходимость аккуратного документирования кода

Язык программирования Java

- Относится к объектно-ориентированным языкам программирования



Write once, run everywhere!

История развития

- 1991 The Green Project, язык Oak



<https://www.youtube.com/watch?v=1CsTH9S79qI&feature=youtu.be>

- 1994 Oak Java

История развития

1996 Java Development Kit 1.0

1997 JDK 1.1

1998 J2SE 1.2

2000 J2SE 1.3

2002 J2SE 1.4

2004 J2SE 5.0

2006 Java SE 6

2011 Java SE 7

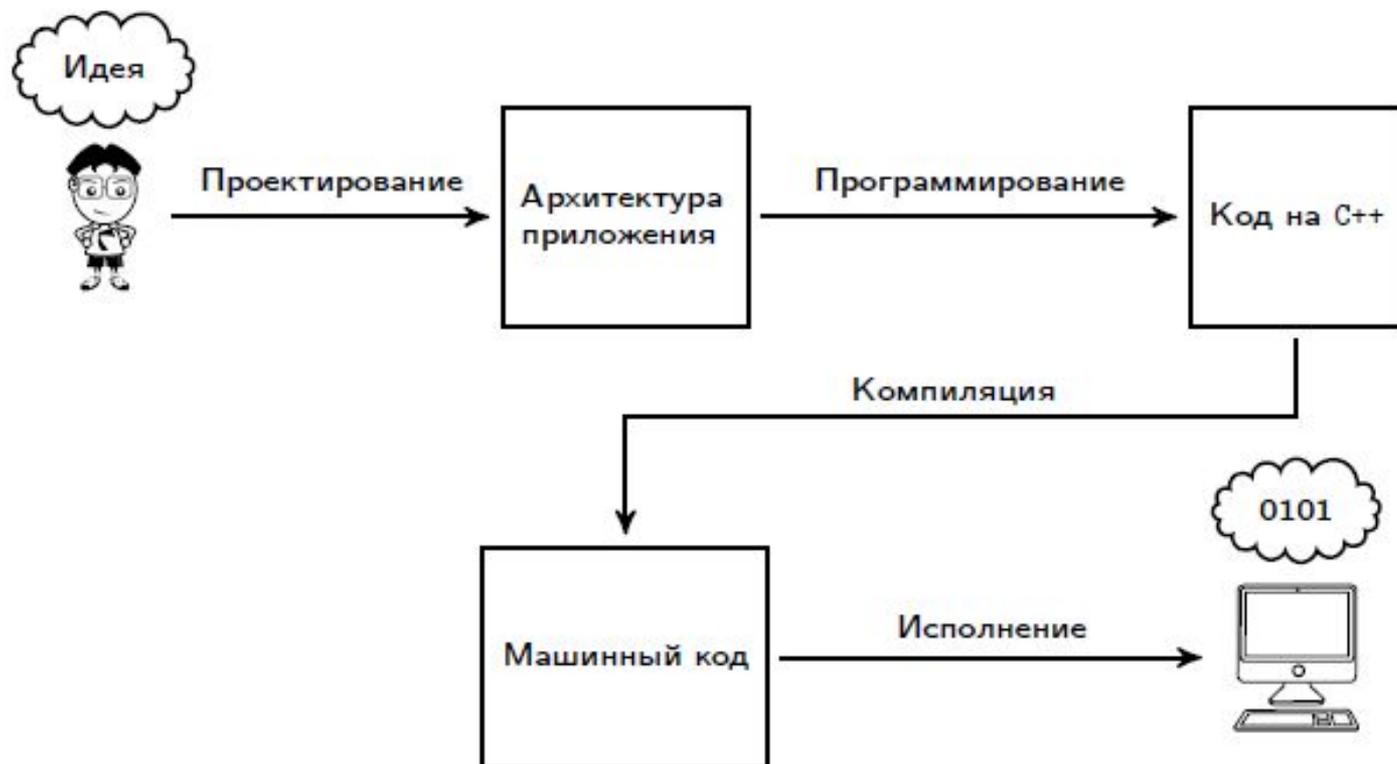
2014 Java SE 8



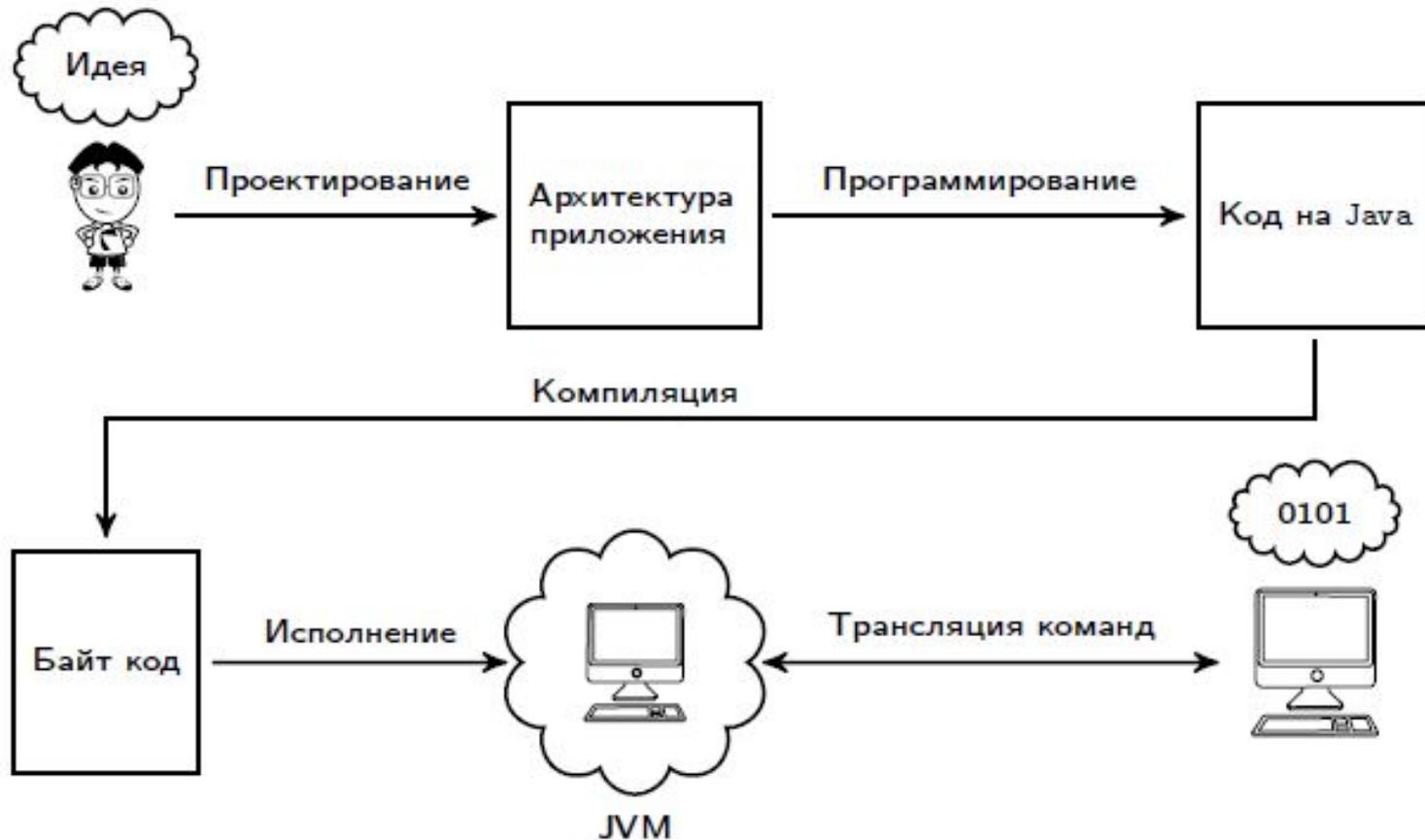
Особенности Java

- Безопасность
- Переносимость
- Байт код
- Java Virtual Machine

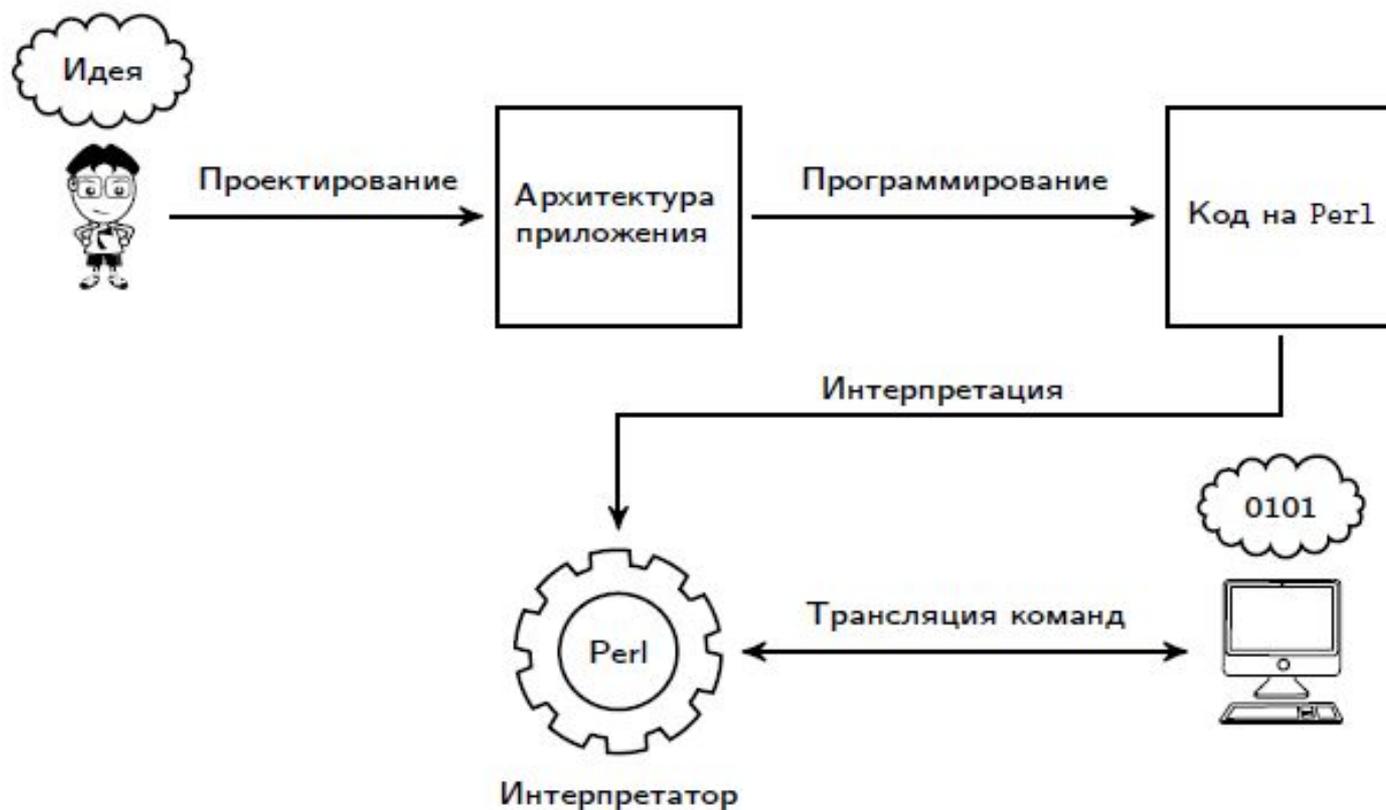
Что такое компиляция?



Что такое компиляция?

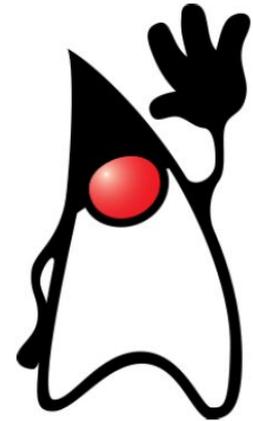


Что такое интерпретация?



Hello, world!

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
    }  
}
```



Java Development Kit

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Запуск из cmd*

- 1. откроем cmd
- 2. чтобы прописать путь к основным директивам JDK нужно обновить переменную окружения PATH, добавим в нее соответствующий путь:
 - set PATH=%PATH%;C:\Program Files\Java\jdk1.8.0_31\bin
- 3. файл должен располагаться в той папке, в которой мы находимся, если необходимо изменить, используйте команду cd
- 4. компиляция: javac HelloWorld.java
- 5. проверим, что появились файлы .class: dir
- 6. если интересно, можем посмотреть на инструкции байт-кода: javap -v HelloWorld.class
- 7. Запуск на выполнение: java HelloWorld

Среды разработки IDE

#9432 +(1067)-

```
<omg> Добрый день. Подскажите пожалуйста хорошую среду разработки для Java.  
<lunxu> среда на следующей неделе будет очень хорошей для разработки на Java
```

- <http://www.eclipse.org/>  eclipse
- <https://netbeans.org/>  **NetBeans**
- <https://www.jetbrains.com/idea/>  IntelliJ**IDEA**