



Урок 1

Базовые понятия, введение в ООП

Agenda

- Введение в платформу Java
- Настройка среды. Пример программы, запуск и выполнение
- Среда разработки (IDE), системы управления версиями (GIT)
- Знакомство со средой разработки Idea v14. Пример программы
- Классы и объекты, поля и методы
- Использование `package`, `import`
- Основные принципы ООП

Настройка среды

Установка JDK

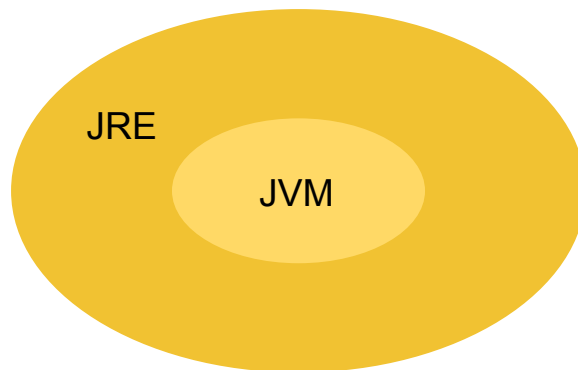
<http://www.oracle.com/technetwork/java/javase/downloads>

Введение в платформу Java



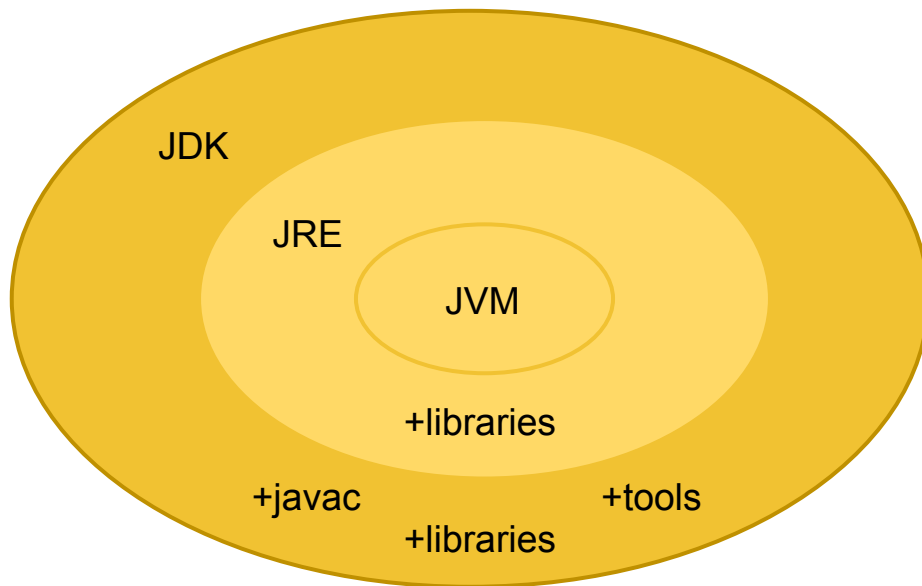
JVM (Java Virtual Machine) - виртуальная машина Java

Введение в платформу Java



JRE (Java Runtime Environment) - среда выполнения Java

Введение в платформу Java



JDK (Java Development Kit) - комплект разработчика Java

Настройка среды

Установка IntelliJ IDEA

<https://www.jetbrains.com/idea/download>

Среды разработки (IDE)

IDE (Integrated development environment) - интегрированная среда разработки

- NetBeans (netbeans.org) - Oracle, ex. Sun
- Eclipse (eclipse.org) - IBM
- IntelliJ IDEA (jetbrains.com/**idea**) - JetBrains



NetBeans



IntelliJIDEA



DonRiver

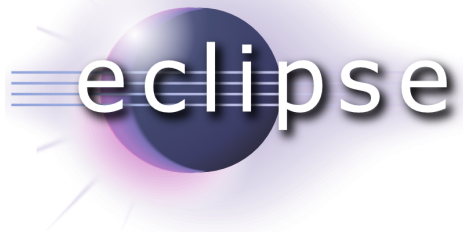
Среды разработки (IDE)

IDE (Integrated development environment) - интегрированная среда разработки

- NetBeans (netbeans.org) - Oracle, ex. Sun
- Eclipse (eclipse.org) - IBM
- IntelliJ IDEA (jetbrains.com/idea) - JetBrains



NetBeans



IntelliJIDEA



DonRiver

Настройка среды

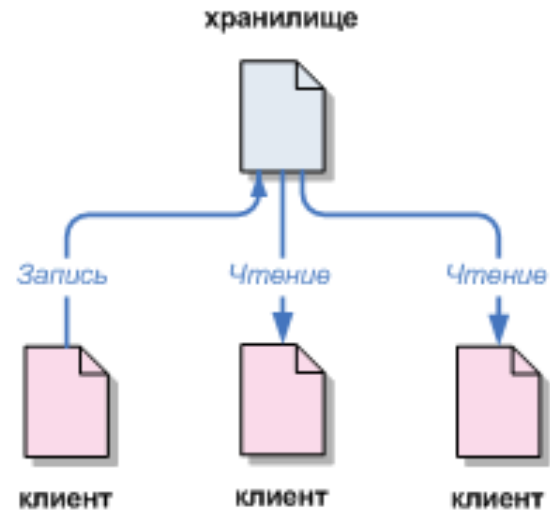
Установка GIT

<https://git-scm.com/downloads>

Системы управления версиями

Свойства

- Атомарность
- Блокировка файла
- Слияние изменений (merging)
- Тэгирование, ветвление

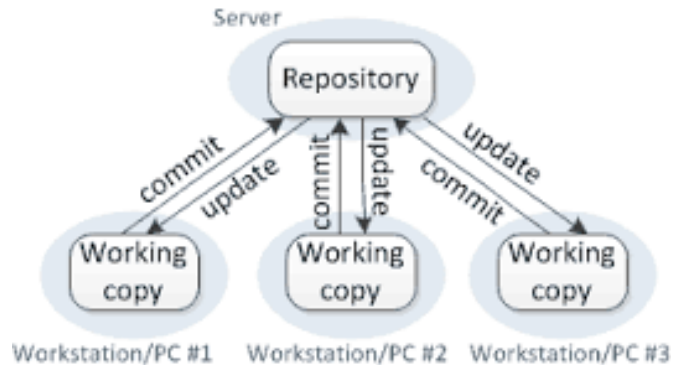


Системы управления версиями

Централизованные

- Subversion
- CVS

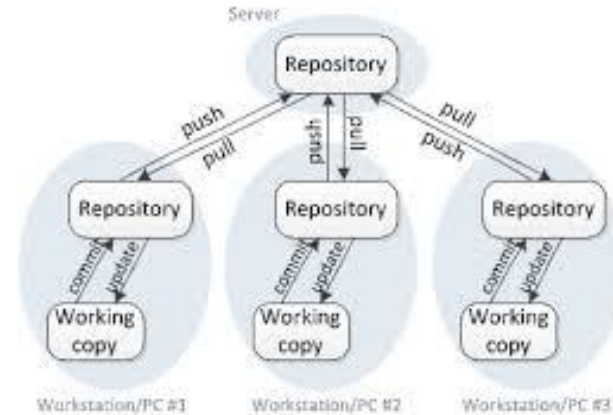
Centralized version control



Распределенные

- Git
- Mercurial

Distributed version control

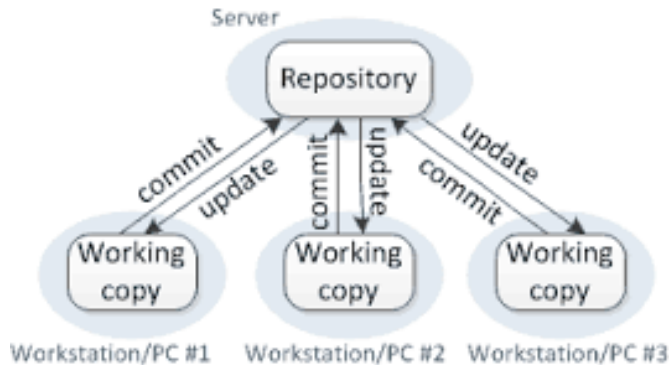


Системы управления версиями

Централизованные

- Subversion
- CVS

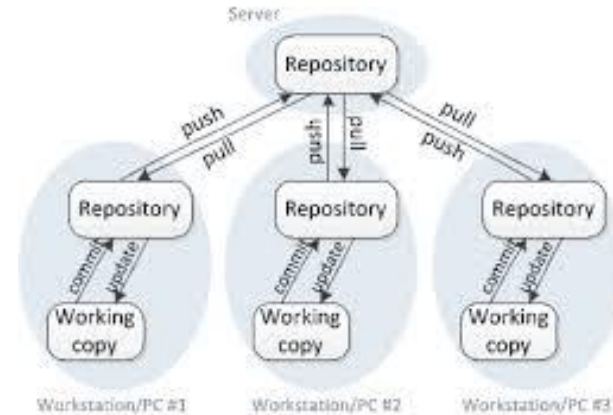
Centralized version control



Распределенные

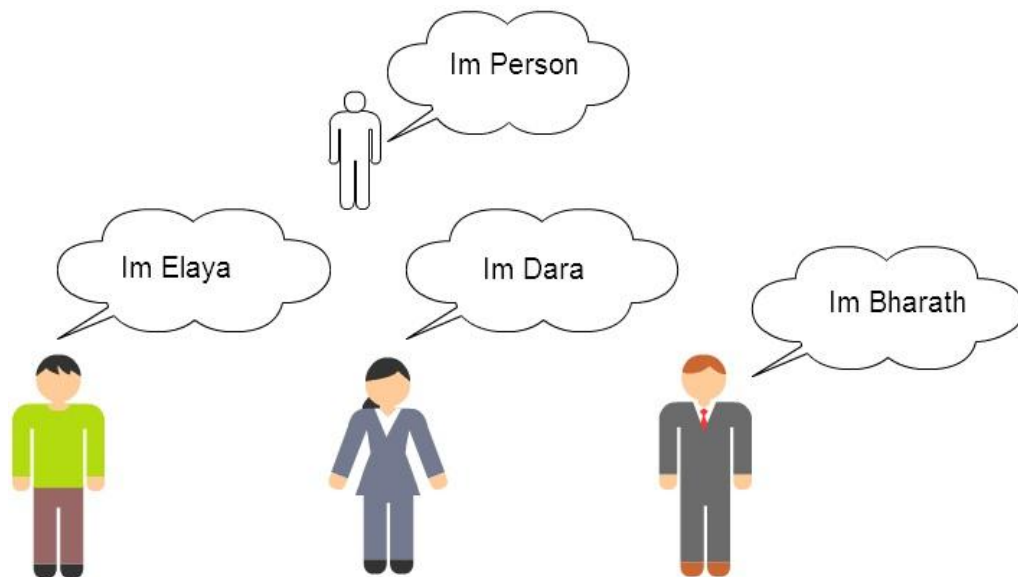
- Git
- Mercurial

Distributed version control



Классы и объекты, поля и методы

- Класс - шаблон для создания объекта
- Объект - экземпляр класса



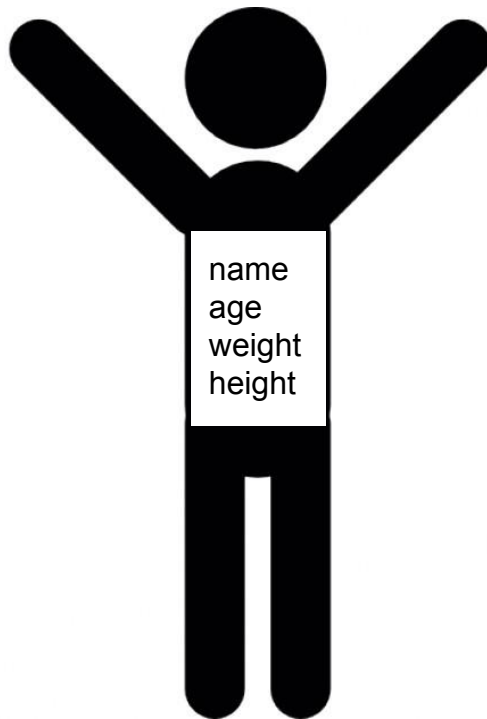
Классы и объекты, поля и методы

- Поля (переменные) - данные, содержащиеся в классе



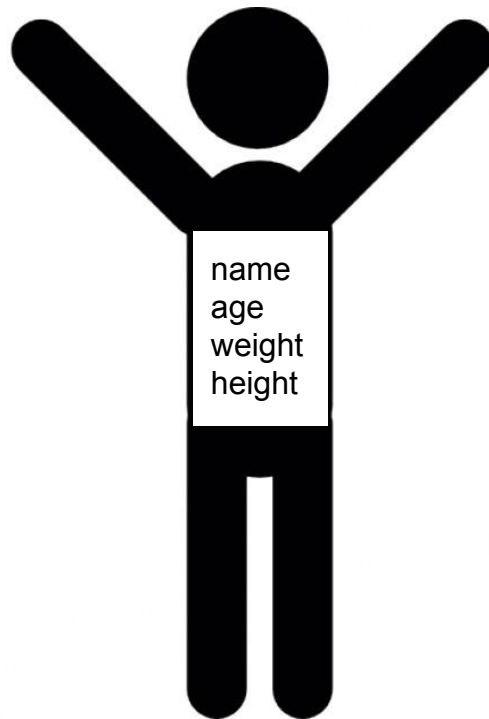
Классы и объекты, поля и методы

- Поля (переменные) - данные, содержащиеся в классе



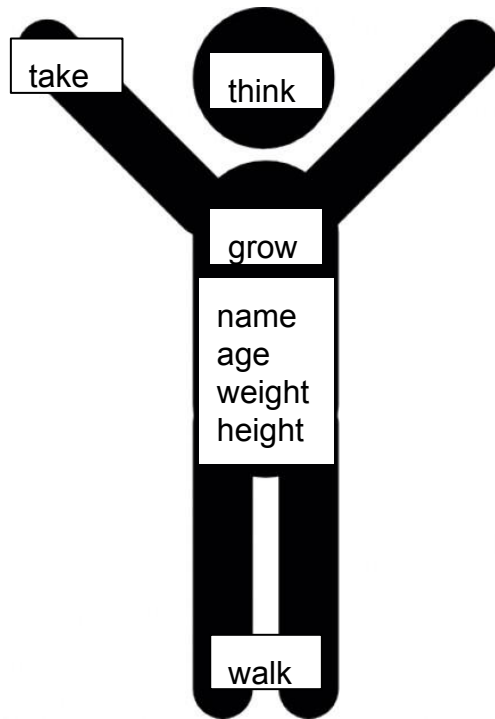
Классы и объекты, поля и методы

- Поля (переменные) - данные, содержащиеся в классе
- Методы - функции, содержащие код доступа к данным



Классы и объекты, поля и методы

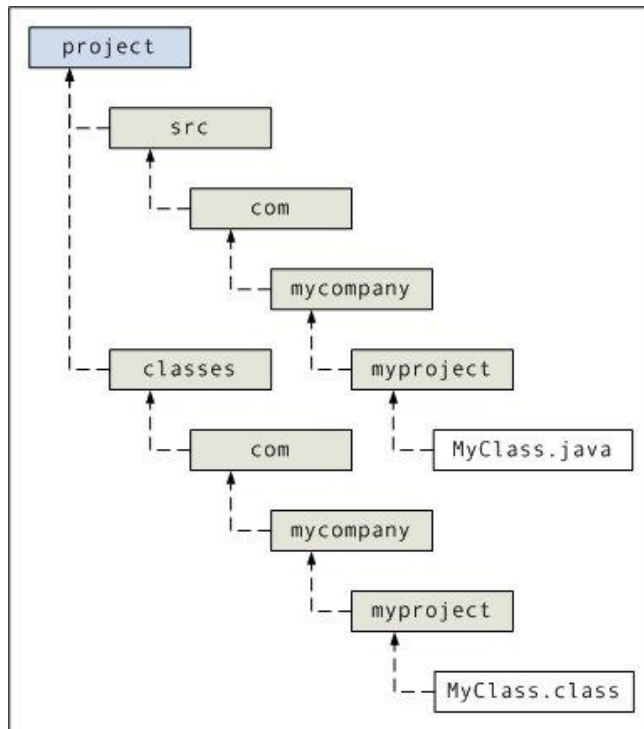
- Поля (переменные) - данные, содержащиеся в классе
- Методы - функции, содержащие код доступа к данным



Использование package, import

- package - определяет пространство имен, в котором хранятся классы

com.mycompany.myproject



Использование package, import

- package - определяет пространство имен (пакет), в котором хранятся классы
- import - вспомогательный оператор для получения доступа к классам, находящимся в пакета

Использование package, import

- package - определяет пространство имен (пакет), в котором хранятся классы
- import - вспомогательный оператор для получения доступа к классам, находящимся в пакета

```
com.example1.Person  
com.example2.Car  
com.example3.House
```

→

```
package com.example4;  
  
import com.example1.Person;  
import com.example2.Car;  
import com.example3.House;  
  
public class City{  
    // code  
}
```

Основные принципы ООП

- Абстракция
- Инкапсуляция
- Наследование
- Полиморфизм

Основные принципы ООП

Абстракция - это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция — это набор всех таких характеристик (Википедия)

Основные принципы ООП

Абстракция - это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция — это набор всех таких характеристик (Википедия)



risovach.ru

Основные принципы ООП

Инкапсуляция - это свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе, и скрыть детали реализации от пользователя. (Википедия)

Основные принципы ООП

Наследование - это свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью. (Википедия)

Основные принципы ООП

Полиморфизм - это свойство системы использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта. (Википедия)

Home work

- определить иерархию классов (минимум 3-х уровневую)
- ознакомиться с Oracle Java Naming Conventions (1-6, 9)
<http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>