

# Java

# Введение

© Составление, Будаев Д.С., Гаврилов А.В., 2013

**Лекция 1**

**NetCracker®**

**УНЦ «Инфоком»  
Самара  
2013**

# План лекции

---

- История языка Java и его особенности
- Объектно-ориентированное программирование, основные понятия
- Пакеты в Java
- Правила именования

# Некоторые факты о Java

- 1 миллиард 100 миллионов компьютеров!!!
- 930 миллионов загрузок JRE каждый год
  - JDK 1.1 загружен 220 000 раз за 3 недели!
- Около 9 миллионов разработчиков
- Все наиболее значимые отрасли
- Огромный список разных типов устройств
  - принтеры, веб-камеры, авто-системы, навигаторы, терминалы, телефоны, TV-приставки, плееры, другая техника.

# Почему Java?

- Простой язык
- Объектная ориентированность
- Устойчивость, предсказуемость
- Поддержка многопоточности
- Архитектурная независимость
- Интерпретируемость и эффективность
- Распределенная технология

# Почему Java?

- Проверена, расширена и признана
- Огромное сообщество разработчиков
- Принцип "write once, run anywhere" (WORA)
- Широкий спектр решаемых задач
- Разнообразный набор библиотек
- Постоянное развитие технологии

# Предыстория Java

- 90-е годы, бурное развитие WWW
- Скандальная история внутри Sun
  - Скотт МакНили (Scott McNealy), CEO
  - Заявление Патрика Нотона (Patrick Naughton)
    - множество средств разработки и архитектур
    - поддержка от Билла Джоя и Джеймса Гослинга
  - Брейнсторминг, идея нового проекта
    - главное – потребитель
    - аппаратно-программная платформа
    - результат – удобное и простое устройство

# Предыстория Java

- Старт проекта Green (1991)
  - Патрик Нотон, Джеймс Гослинг, Майк Шеридан
  - Идея Гослинга об "универсальном пульте"
  - Модификации Гослингом языка C++
  - Начало работ над ОаК, "технология молотка"
- Первая демонстрация (08.1991)



# Предыстория Java

---

Идеи, заложенные в ОаК, проект Green:

- Надежность и механизмы безопасности
- Работа на разных типах устройств
- Объектная ориентация
- Объекты, доступные по сети

# Предыстория Java

---

Развитие интерактивного TV (1992-1994)

- Mitsubishi, France Telecom, Dolby Labs
- Создание компании FirstPerson
- Конкурс Time Warner, TV-приставки
- Трехмесячные переговоры с 3DO
- Отказ от идеи интерактивного TV

# Предыстория Java

- Разработка HTML, Европейский институт физики частиц (CERN), 1991
- Первая версия браузера Mosaic, 04.1993
- Бесплатность для конечного пользователя
- Создание нового бизнес-плана для OaK
- Адаптация OaK для задач WWW
- Браузер WebRunner, апплеты, 09.1994

# Предыстория Java

- Демонстрация возможностей, 1994-1995
- Переименование ОаК в Java
  - Neon, Lyric, Pepper, Silk
- Переименование WebRunner в HotJava
- Лавинообразный рост загрузок
- Конференция SunWorld, 1995
  - соглашение с Netscape (Джеймс Кларк)

# Предыстория Java

- 1991  
Начало работ над проектом Green, языком Oak
- 1993  
Работы в области интерактивного TV; приставки  
Появление браузера Mosaic
- 1994  
Браузер WebRunner, переименован в HotJava
- 1995  
Официальное представление Java  
Включение в Netscape Navigator 2.0

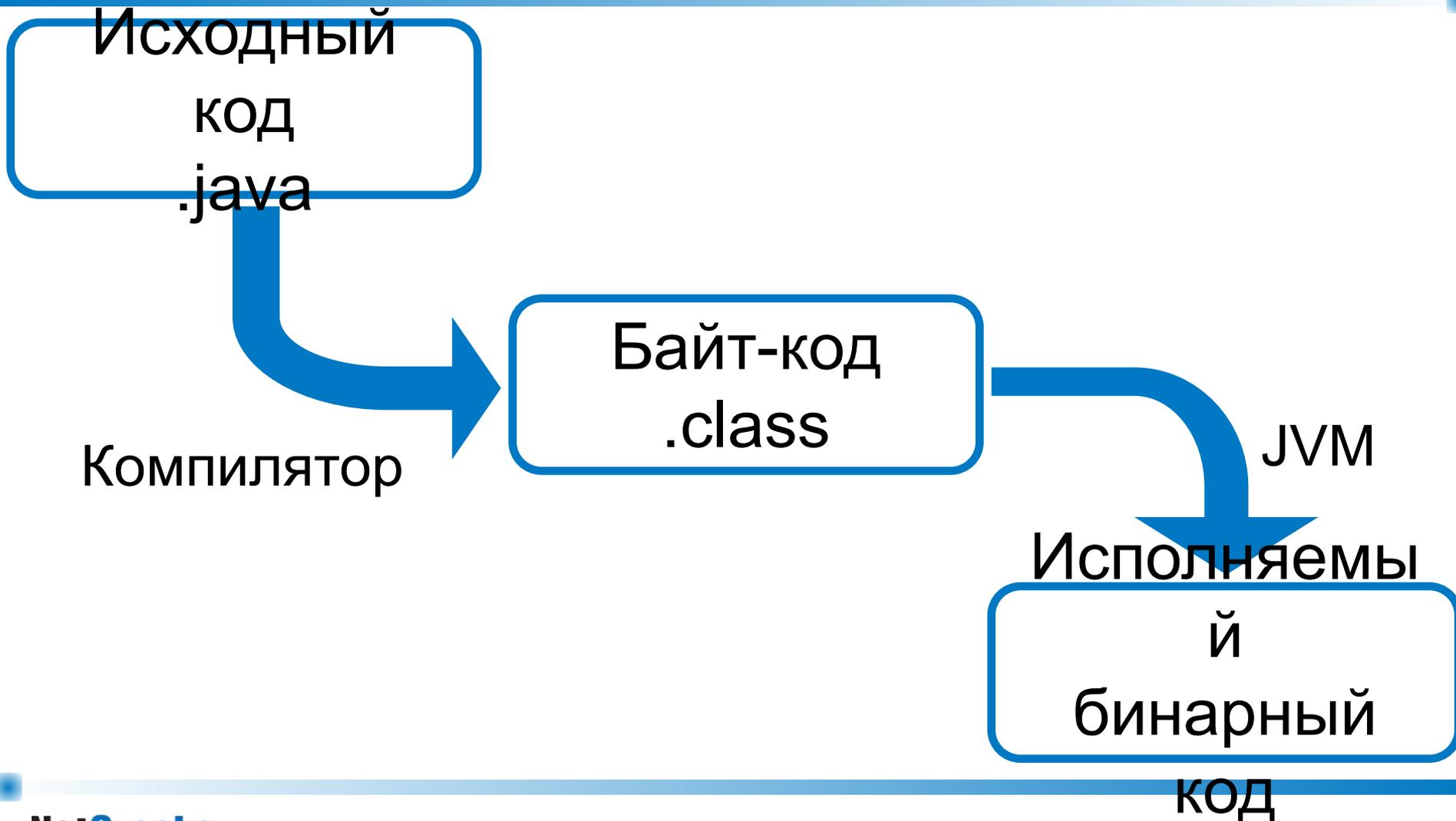
# История Java

- 1996 – JDK 1.0 (JLS, JVM, JDK)
- 1997 – JDK 1.1 (JIT, JavaBeans, JDBC, RMI)
- 1998 – JDK 1.2 (изменения языка, policy/permission, JFC, ...)
- 1999 – разделение развития
  - Java 2 Platform, Standard Edition (J2SE, JavaSE)
  - Java 2 Platform, Enterprise Edition (J2EE, JavaEE)
  - Java 2 Platform, Micro Edition (J2ME, JavaME)
- 2000 – JDK 1.3 (HotSpot (JIT) в составе JVM, ...)
- 2002 – JDK 1.4 (новое API)
- 2004 – JDK 1.5 (изменения языка)
- 2006 – JDK 1.6 (скриптовые языки, работа с базами данных...)
- 2011 – JDK 1.7 (изменения языка...)

# Java платформа

- Множество различных аппаратных систем
  - Intel x86, Sun SPARC, PowerPC и др.
- Множество разных программных систем
  - MS Windows, Sun Solaris, Linux, Mac OS и др.
- Потребность в одинаковом функционале на различных платформах
- Java Virtual Machine (JVM), универсальность
  - Исходный код открыт с 1999 г.

# Разработка и запуск



# Особенности Java

- Строгая типизация
- Кросс-платформенность
- Объектная ориентированность
- Встроенная модель безопасности
- Фокус на распределенные приложения
- Динамичность, но и легкость в освоении
- Механизм автоматической сборки мусора

# Именованние установочных файлов

- Старый вариант  
`jdk-1_5_0_08-windows-i586-p.exe`
  - 1 – глобальная версия языка
  - 5 – номер версии языка
  - 0 – номер подверсии
  - 08 – номер модификации
  - windows-i586 – платформа
- Новый вариант  
`jdk-6u14-windows-i586.exe`
  - 6 – номер версии языка
  - 14 – номер модификации
  - windows-i586 – платформа

# Модели организации программного кода

- Алгоритмическая модель
  - естественна для вычислительных задач
  - описывает последовательность действий
  - фактический отказ от оператора `goto`
- Процедурная модель
  - выделение повторяющихся кусков кода
- Структурное программирование (70-80гг)
  - декомпозиция на совокупность процедур

# ООП подход

- Основан на представлении предметов и явлений предметной области задачи в виде множества программных объектов
- Объекты имеют состояния и обладают соответствующим поведением
- Объекты взаимодействуют друг с другом

# Развитие подходов

## Инструкции

- Сплошные
- С операторами
- Процедуры
- Модули

## Данные

- Ячейка памяти
- Переменные
- Массивы
- Объединения

## ООП

Объединение данных и методов их обработки

# Этапы программного решения задачи

- Создание модели, определение данных для предстоящей обработки
- Разработка алгоритма: определение операций над данными и последовательности шагов по преобразованию текущего состояния модели в следующее
- Формулировка модели и алгоритма на языке программирования

# Объекты и классы

## Объект

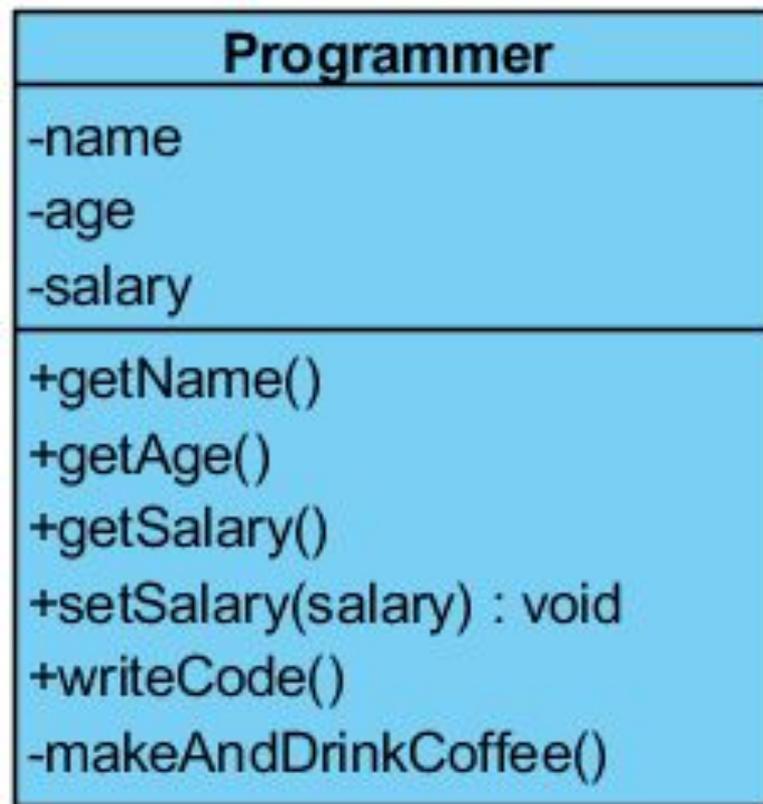
- Состояние
- Поведение
- Уникальность
- Время жизни

## Класс

- Объекты имеют одинаковый набор свойств
- Объекты имеют общее поведение

# Класс

- Шаблон поведения объектов данного типа с некоторыми параметрами
- Название, свойства, конструктор, методы



# Объект

- Состояние
- Поведение
- Уникальность
- Время жизни

```
javaProgrammer : Programmer
```

```
age = 25
```

```
name = Joe
```

```
salary = 100500
```

# Основные принципы

## ■ Инкапсуляция

объединение данных и методов их обработки в одну сущность, приводящее к сокрытию реализации класса и отделению его внутреннего представления от внешнего

## ■ Наследование

отношение между классами, при котором один класс использует структуру или поведение другого (одиночное наследование) или других (множественное наследование) классов

## ■ Полиморфизм

способность объекта соответствовать во время выполнения двум или более возможным типам

# Инкапсуляция

- Данные и методы объекта могут обладать разной степенью доступности
- Открытые методы составляют внешний интерфейс объекта
- Контроль допустимости значений полей
- Модульная схема взаимодействия отдельных компонентов системы
- Легче отлаживать и поддерживать

# Наследование

- Отношение между классами, использование поведения другого класса
- Вводит иерархию наследования классов
- Подклассы дополняют и/или переопределяют структуру и поведение
- Уход от повторного написания кода
- Более эффективный и гибкий код

# Полиморфизм

- "Имеющий много форм" (лат.)
- Тип – класс или интерфейс
- Иерархия различных типов
- Возможность использования объектов более общего полиморфного типа
- Полиморфный вызов разных по реализации, но одноименных методов

# Отношения между классами

- **Ассоциация**

Объекты классов вступают во взаимодействие между собой

- **Агрегация**

Объекты одного класса входят в объекты другого, при этом не обладая самостоятельностью

- **Композиция**

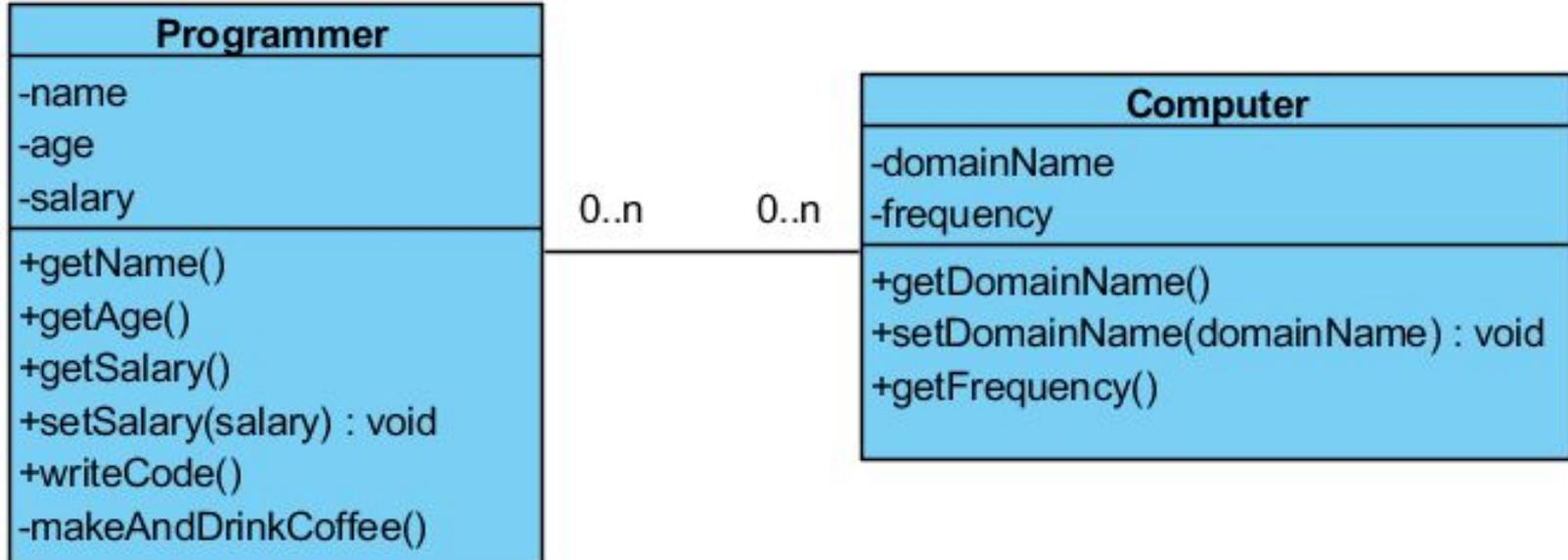
Объекты разных классов образуют целое, оставаясь самостоятельными

- **Класс-метакласс**

Экземплярами класса являются классы

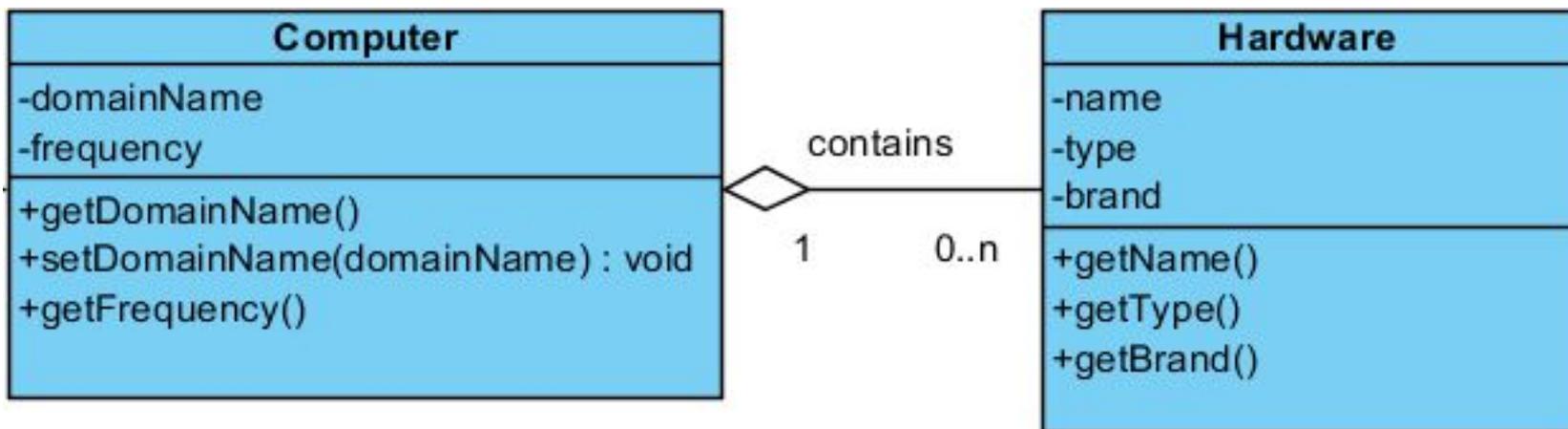
# Ассоциация

- Объект(ы) одного типа ссылаются на объект(ы) другого типа
- Отношение не носит характера владения



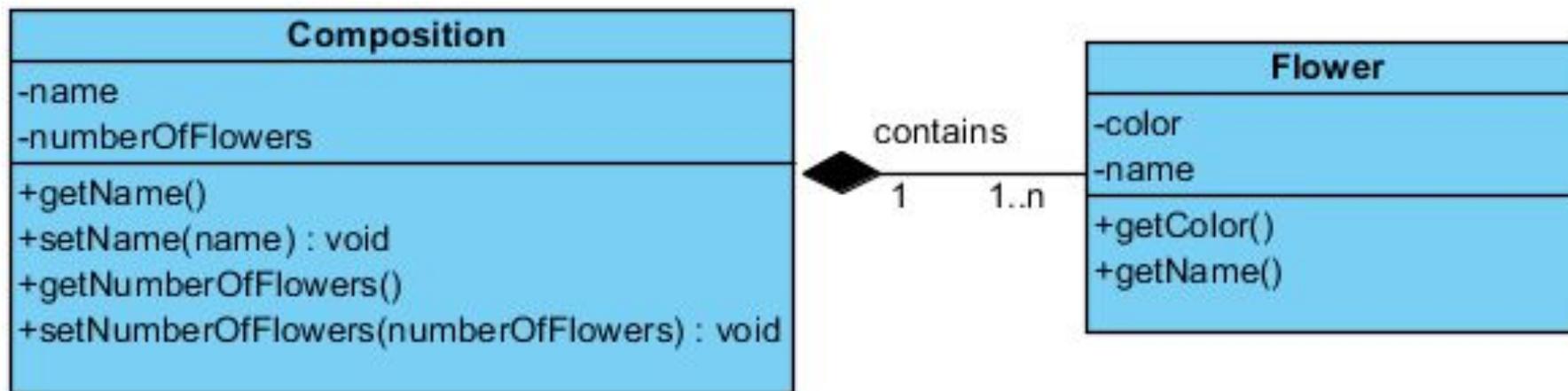
# Агрегация

- Отношение между классами типа "содержит" или "состоит из"
- При исчезновении своих составных частей объект-агрегат **продолжает** существовать



# Композиция

- Объекты разных классов образуют целое, оставаясь самостоятельными
- При исчезновении своих составных частей объект **перестает** существовать



# Класс-метакласс

- Класс можно рассматривать как объект со свойствами и методами
- Можно создать шаблон, которому должны удовлетворять все создаваемые классы
- "Инструкция по созданию инструкций"
- Объекты порождаются от классов, а классы – от метаклассов
- Метаклассы – являются классами

# Достоинства ООП

- Упрощение разработки и сохранение СОВМЕСТИМОСТИ
  - Модульность и локализация сущностей, возможность независимых модификаций компонентов, повторное их использование
- Возможность создания расширяемых систем
  - Обработка разнородных структур, наследование и изменение поведения на этапе выполнения
- Повышение надежности программного обеспечения

# Недостатки ООП

- Неэффективность на этапе выполнения
- Неэффективность в смысле распределения памяти
- Излишняя избыточность
- Психологическая сложность проектирования
- Техническая сложность проектирования и документирования

# Объектный язык Java

---

- **Все** сущности в Java являются объектами, классами либо интерфейсами
- **Строгая** реализация инкапсуляции
- Реализовано одиночное наследование от класса и **множественное** от интерфейсов

Спасибо за внимание!

# Дополнительные источники

- Арнолд, К. Язык программирования Java [Текст] / Кен Арнолд, Джеймс Гослинг, Дэвид Холмс. – М. : Издательский дом «Вильямс», 2001. – 624 с.
- Вязовик, Н.А. Программирование на Java. Курс лекций [Текст] / Н.А. Вязовик. – М. : Интернет-университет информационных технологий, 2003. – 592 с.
- Хорстманн, К. Java 2. Библиотека профессионала. Том 1. Основы [Текст] / Кей Хорстманн, Гари Корнелл. – М. : Издательский дом «Вильямс», 2010 г. – 816 с.
- Эккель, Б. Философия Java [Текст] / Брюс Эккель. – СПб. : Питер, 2011. – 640 с.
- JavaSE at a Glance [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/technetwork/java/javase/overview/index.html>, дата доступа: 21.10.2011.
- JavaSE APIs & Documentation [Электронный ресурс]. – Режим доступа: <http://www.oracle.com/technetwork/java/javase/documentation/api-jsp-136079.html>, дата доступа: 21.10.2011.