Java

Введение в ООП

План занятия

- Классы в Java
- ООП
  - Наследование
  - Инкапсуляция
  - Полиморфизм
  - Абстракция

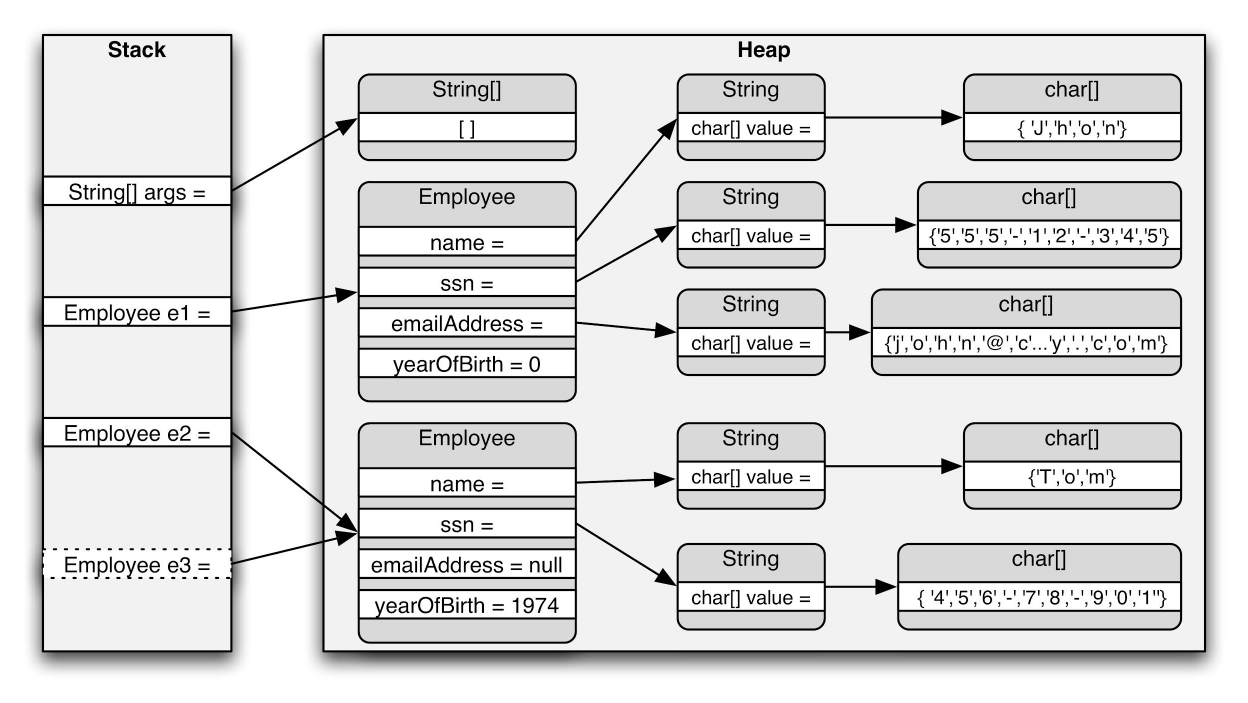
Классы в Java

- Классы описывают объекты реального мира
- Класс состоит из полей и методов и конструкторов
- Экземпляр класса описывает конкретный объект
- Создаются оператором new

Модификатор static

- Означает принадлежность к классу, а не экземпляру
- Методы и поля могут быть статическими
- Статические блоки

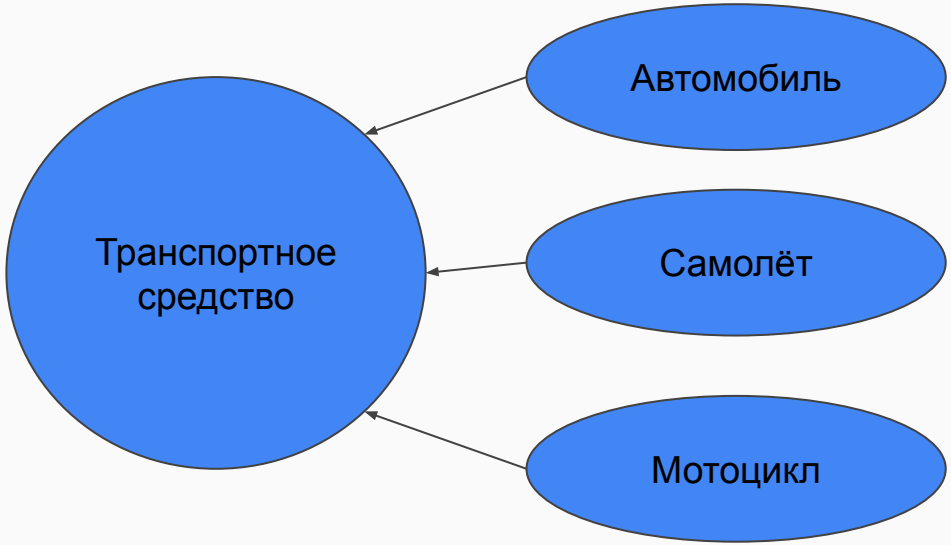
Пример структуры Java объекта



Объектно-ориентированное программирование

- Парадигма программирования
- Основные принципы:
  - Наследование
  - Инкапсуляция
  - Полиморфизм
  - Абстракция

Наследование



Наследование

- Ключевое слово - extends
- Позволяет переиспользовать часть кода
- Позволяет выделять общее поведение
- Потомок может использоваться вместо родителя
- Потомок может переопределять поведение родителя (@Override)
- В Java нет множественного наследования

Object

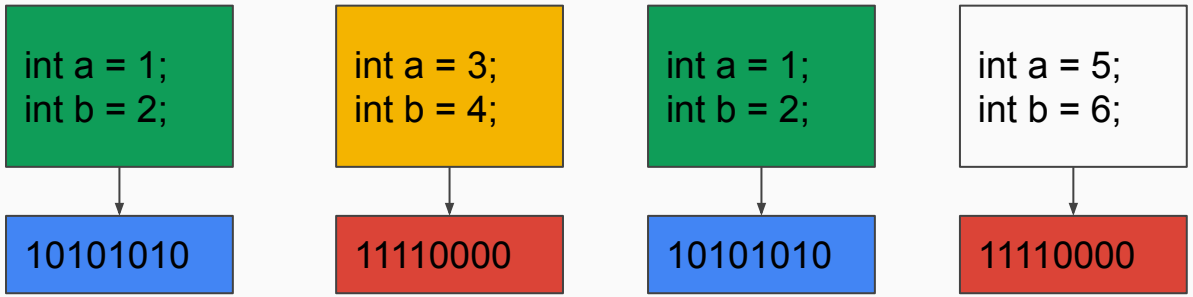
- Все объекты наследуются от java.lang.Object
- В Java всё является объектом
- Исключение - примитивные типы
- int vs Integer

Основные методы Object

- hashCode()
- equals(Object o)
- toString()

hashCode()

- хеш - число
- Функция в которую можно передать данные и получить от них число
- Используется в коллекциях



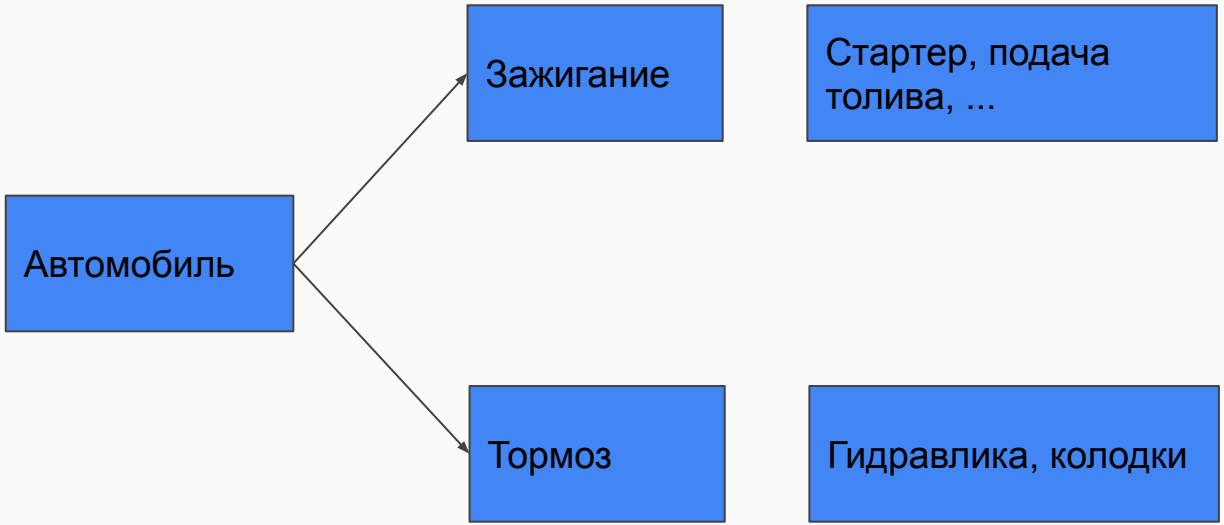
equals(Object o)

- == сравнивает ссылки!
- По умолчанию сравнивает ссылки
- Если equals(..) = true то hashCode() равны
- Если hashCode() равны то equals(..) не обязательно true

toString()

- По умолчанию имя класса + hashCode()
- Переопределяется для удобного вывода и отладки

Инкапсуляция



Инкапсуляция

- Контроль доступа
  - Константы, ограничение видимости
- Контроль целостности
  - Логика работы с внутренними данными (полями)
- Замена реализации
- Контракты для разработчиков
- Пакеты в Java
  - Не иерархичны

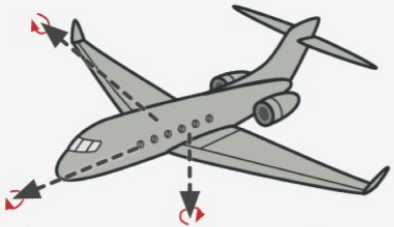
Инкапсуляция (видимость)

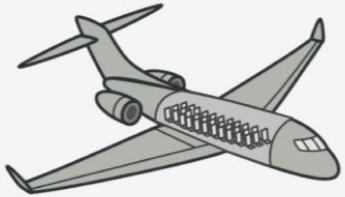
	default	private	protected	public
Тот же класс	Да	Да	Да	Да
Подкласс в том же пакете	Да	Нет	Да	Да
Другой класс, тот же пакет	Да	Нет	Да	Да
Подкласс в другом пакете	Нет	Нет	Да	Да
Другой класс, другой пакет	Нет	Нет	Нет	Да

Полиморфизм

- Дочерний класс может быть использован везде, где используется родительский
- Если дочерний класс приведен к родительскому, то доступны только методы родительского класса (по типу ссылки)
- Вызывается реализация по реальному типу объекта (@Override)

Абстракция





Airplane
- speed - altitude - rollAngle - pitchAngle - yawAngle
+ fly()

Airplane
- seats
+ reserveSeat(n)

Разные модели одного и того же реального объекта.

Интерфейсы в Java

- Определяет, что можно сделать с классом
- Не определяет, как это делать
- Класс может реализовывать несколько интерфейсов
- Абстракция от реализации
- Обобщение по свойству
- Реализацию можно изменить
- Контракты (API)