Лекция 2

Синтаксис

- Переменная должна быть инициализирована до использования в выражении
- Константы – final, после инициализации нельзя изменить
- Полям класса присуждаются значения по умолчанию
- Идентификаторы: начинаются с “_ ; $; (char)” регистро-зависимы
- const и goto зарезервированы но не используются
- Система типов: Value-types (примитивы) и Reference-types (ссылочные или объектные типы)
- Суффикс l, L – long, d, D – double. Типы по умолчанию: целочисленные : int, вещественные: double
- double имеет специальные значения: Infinity, -Infinity, NaN
- bool имеет значения true или false (~~1, 0~~)
- char: одинарные ковычки. Пример: ‘a’, ‘\u03A6’ – символ кодировки Unicode
- String – это класс, а не примитив. Текстовые блоки.
- Преобразования типов:
    1. Неявные без потери точности
    2. Неявные с потерей точности
    3. Явные (сужающие)
- При преобразовании int → float может произойти потеря точности т.к. изменяется способ хранения данных
- Ссылочные типы хранят ссылку на объект, new возвращает ссылку на созданный объект
- Аргументы в методы всегда передаются по значению
    1. Аргументы примитивы – создаются копии объекта
    2. Аргументы ссылки – создаются копии ссылки на тот же объект
- Если типы операторов не совпадают, выполняется неявное расширение типа операторов до общего.
- Короткое замыкание && и ||

- Тернарные оператор: int sign = x >= 0 ? 1 : -1
- instanceof
- switch
- for each
- Метки: можно брейкать внешний цикл при этом все внутренние сразу прерываются

ООП

- First-class citizen
- Состояние класса должно быть всегда корректным. Контроль за состоянием важен
- Поля могут быть инициализированы при определении
- При наличии одного конструктора, конструктор по умолчанию не добавляется
- Инкапсуляция: сокрытие данных → доступ к данным осуществляется через методы
- private, default (package private), protected, public
- Наследование: описание нового класса на основе старого, с целью заимствования функционала
- class Rectangle extends Shape
- Java не поддерживает множественное наследование
- Для запрета наследования от класса его можно пометить модификатором final
- this() and super() - литералы доступа к полям и конструкторам текущего и базового класса. Такие конструкторы должны вызываться только в качестве первой строки в конструкторе. Если обращаемся к полям или методам, то в любом месте, кроме статических областей.
- Полиморфизм: позволяет использовать уникальное поведение объектов с общим предком (базовым классом) без спецификации типов этих объектов
- Overloading: название методов совпадают, но списки параметров различаются
- Виртуальные методы – вызывается метод, соотв. Конкретному типу объекта в процессе выполнения
- Гетерогенные коллекции
- instanceof: является ли производным классом или объектом типа
- Абстрагирование: Выделение значимых характеристик классов и исключение незначимых.

- Абстрактные классы: экземпляры нельзя создавать, наследники абстрактного класса должны переопредлить все его абстрактные методы
- Интерфейсы. Duck implements Swimmable. При этом в классе дак нужно обязательно реализовать все методы интерфейса
- default можно указать в интерфейсе, чтобы не реализовывать его в каждом классе