Algorithms and data structures

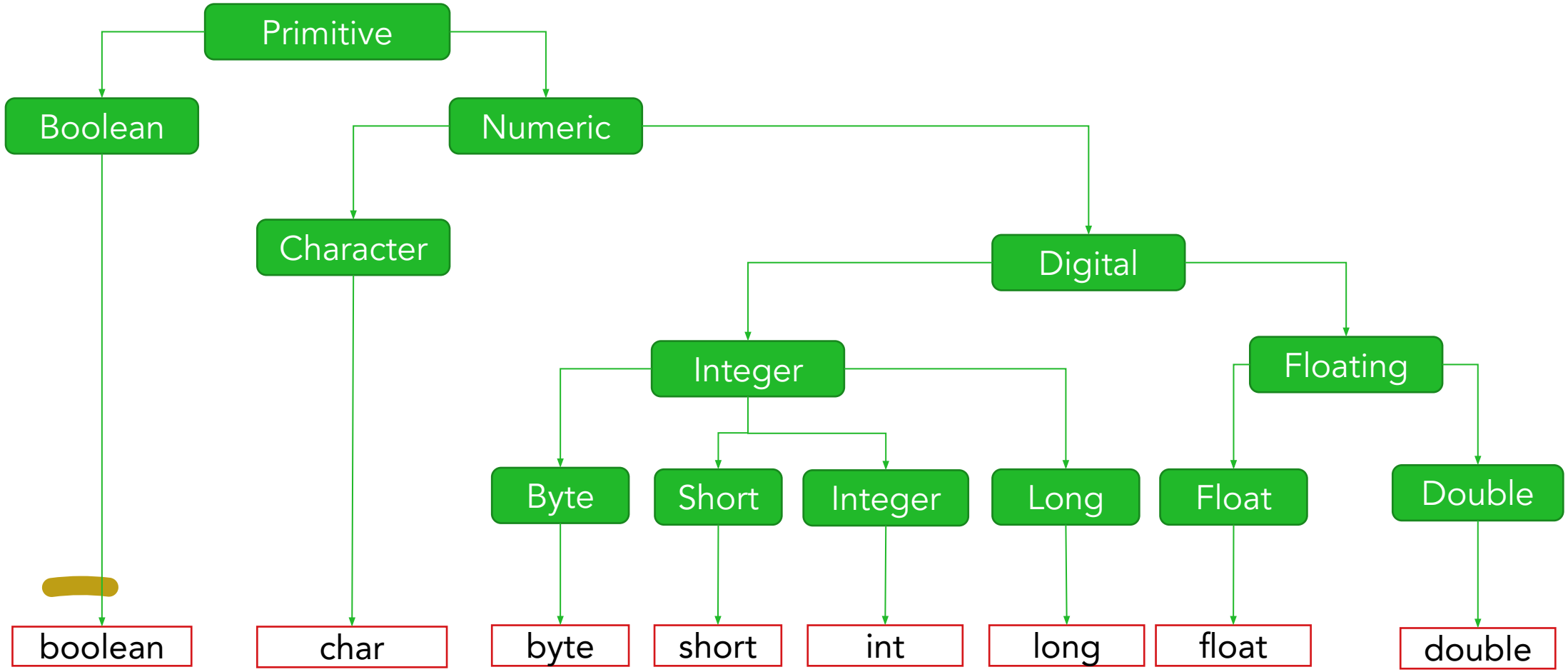
lecture #4. Reference Data Types. Types of References in Java
Mentor: Aleks Reingand

lecture #4. Reference Data Types. Types of References in Java

- Reference Data Types
  - primitive
  - Reference
    - properties and definitions
    - Memory allocation and transformation of primitives
      - autoboxing
      - unboxing
    - Compare
    - Copying
- Types of References in Java
  - 4 types of references

Primitive

boolean, char, int, short, byte, long, float и double



Reference data types

Array - предоставляет структуру данных фиксированного размера, в которой хранятся элементы одного типа.

Class - набор инструкций, описывающих содержание объекта.

Interface - реализуется классами Java.

String - представляет строки символов.

Enum - особый вид класса, типобезопасный. Каждый элемент внутри enum является экземпляром этого Enum.

properties and definitions

- Не предопределено, кроме String.
- Все ссылочные типы начинаются с заглавной буквы.
- Используется как вызывающий или для вызова методов.
- Может быть null.
- JVM по умолчанию выделяет 8 bytes для каждой ссылочной переменной.

Memory allocation and transformation of primitives

Ключевое слово new используется для создания экземпляра класса

Выделяется память для нового объекта, и возвращается ссылка на этот объект (эту память)

Объекты занимают память в пространстве кучи (Heap) Java

Если ссылок на объект нет, память, используемая этим объектом, будет освобождена в процессе сборки мусора.

Вопрос - ???

Преобразование примитивного типа в ссылочный тип называется автоупаковкой (autoboxing)

Преобразование ссылочного типа в примитивный тип называется распаковкой (unboxing)

autoboxing and unboxing

- Для примитивных типов данных существуют классы-оболочки
  - Автоупаковка (autoboxing) относится к преобразованию примитивного значения в объект соответствующего класса-оболочки.
    - Например, преобразование int в класс Integer.
  - Компилятор Java применяет автоупаковку, когда примитивное значение:
    - Передается как параметр методу, который ожидает объект соответствующего класса-оболочки.
    - Присваивается переменной соответствующего класса-оболочки.

  - Распаковка (unboxing) относится к преобразованию объекта типа-оболочки в соответствующее ему примитивное значение.
    - Например, преобразование Integer в int.
  - Компилятор Java применяет распаковку, когда объект класса-оболочки:
    - Передается как параметр методу, который ожидает значение соответствующего примитивного типа.
    - Присваивается переменной соответствующего примитивного типа.

Compare

- Два способа сравнения ссылочных типов:
  - Используя оператор равенства (==)
  - Используя метод Object.equals()

== сравнивает расположение объектов в памяти.

Если адрес памяти (ссылка) обоих объектов один и тот же, объекты равны.

Не сравнивает содержимое объекта.

equals() метод класса Object.

Не переопределенный, также использует оператор равенства (==) для сравнения ссылочного типа.

Для сравнения содержимого объекта требуется переопределить метод.

Copying

Для копирования объекта существует несколько путей:

1. Создаем копию ссылки на объект

```
Book pinokkio = new Book(“Pinokkio", 1950);
pinokkio.print();      // Book Pinokkio
Book miracle = pinokkio;
miracle.setName(“Miracle");
pinokkio.print();      // Book mirackle
```

2. Создаем фактическую копию (создание новой копии) объекта. (clone)

Для реализации клонирования класс Book должен имплементировать интерфейс Cloneable, который определяет метод clone.

Реализация этого метода просто возвращает вызов метода clone для родительского класса - то есть класса Object с преобразованием к типу Book.

3. А есть что-то еще?

Types of References in Java

В Java существует четыре типа ссылок, различающихся по способу их сборки мусора:

1. Сильные (Strong) ссылки
2. Слабые (Weak) ссылки
3. Мягкие (Soft) ссылки
4. Фантомные (Phantom) ссылки

2. Сильные ссылки: это тип/класс ссылочного объекта по умолчанию. Любой объект, имеющий активную сильную ссылку, не подлежит сборке мусора.
3. Слабые ссылки: не являются типом/классом ссылочных объектов по умолчанию, и их следует явно указывать при их использовании.
4. Мягкие ссылки: также не собирается мусором, пока JVM не сильно нуждается в памяти. Объекты очищаются из памяти, когда JVM не хватает памяти.
5. Фантомные ссылки: подлежат сборке мусора. Но перед тем, как удалить их из памяти, JVM помещает их в очередь, называемую «очередью ссылок».