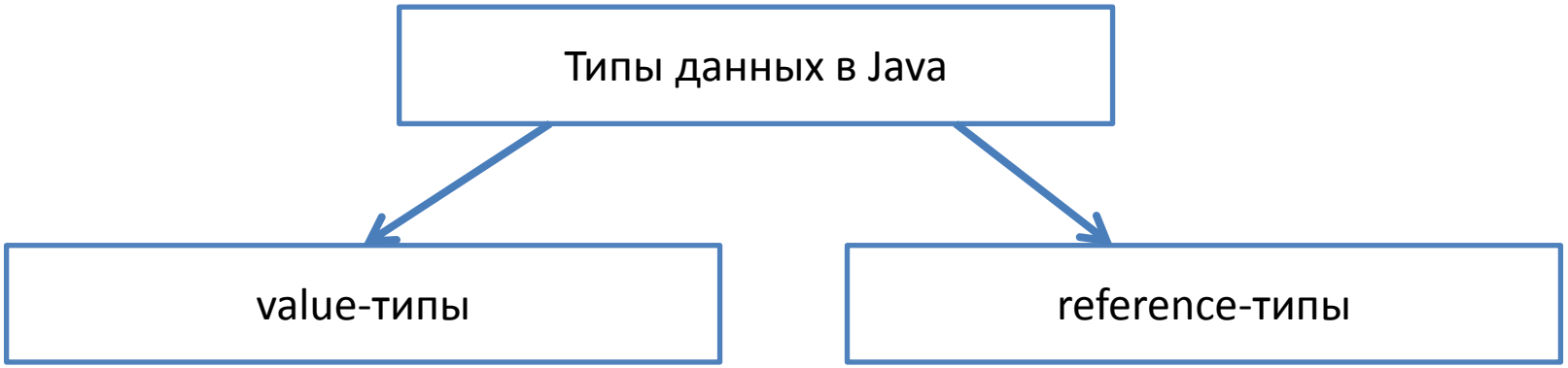
Лекция 8. Примитивные и ссылочные типы. Символьный тип

Типы данных в Java

- Все типы в Java можно разделить на две категории: **value-типы (типы значений)** и **reference-типы (ссылочные типы)**
- Типы из данных категорий ведут себя по-разному
- К value-типам относятся, например числа, а к reference-типам относятся строки

Типы данных в Java

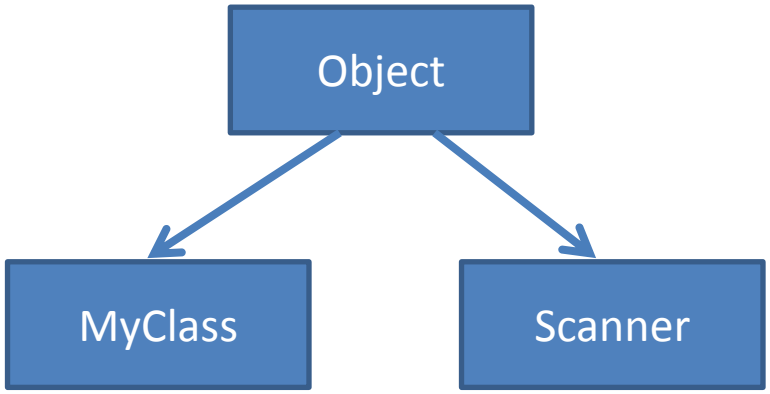


Числовые целые:
byte, short, int, long

Вещественные:
float, double

Логический: boolean

Символьный: char



Все классы наследуются от класса Object

Value-типы

- Их 8 штук
- Все числовые типы:
  byte, short, int, long; float, double
- Логический тип boolean
- Символьный тип char (рассмотрим позже)

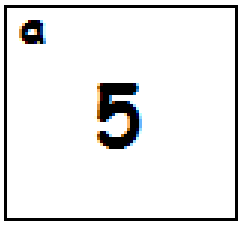
- В Java эти типы называют **примитивными**

Value-типы

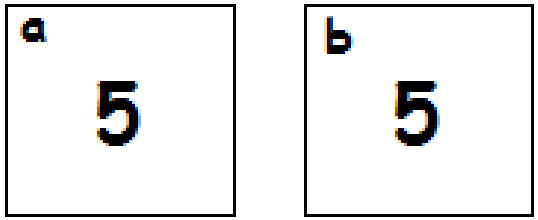
- Переменные value-типов хранят само значение типа
- При присваивании происходит копирование значения
- При передаче аргументов в функции, происходит копирование аргумента

Как работают value-типы

- int a = 5;



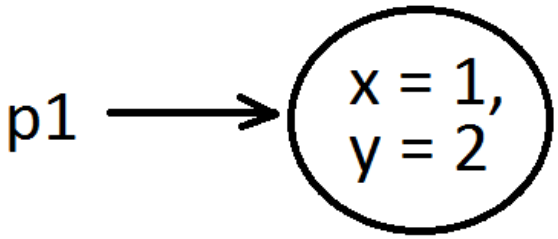
- int b = a;



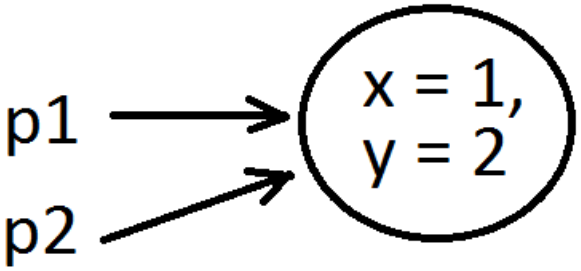
- Если изменить a или b, то это не повлияет на другую переменную

Reference-типы

- Переменные хранят не само значение, а **ссылку** на него (по сути – адрес в памяти)
- `Point p1 = new Point(1, 2);`

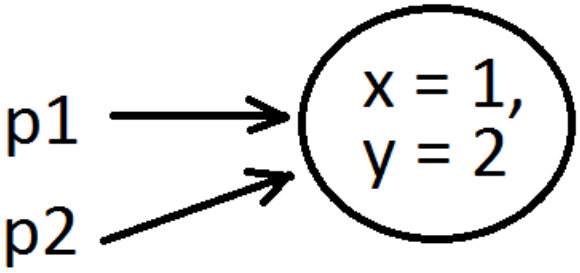


- При присваивании происходит копирование ссылки:
- `Point p2 = p1;`



- В Java все классы являются reference-типами

Reference-типы



- Если изменим объект,
  то все ссылки будут
  указывать на измененный объект
- p2.setX(3);
  System.out.println(p1.getX()); // 3
- При передаче объекта в функцию, происходит копирование ссылки на него
- Зачем нужны ссылки? Чтобы более эффективно работать с памятью. Объекты часто являются большими, и копировать их очень затратно по времени и памяти

Проверка объектов на равенство

- Для объектов нельзя использовать проверку через == и !=
- Для объектов оператор == проверяет, что ссылки указывают на один и тот же объект в памяти или нет
- Аналогично != проверяет, что ссылки указывают на разные объекты
- Чтобы сравнить содержимое объектов, нужно использовать метод equals
- boolean x = o1.equals(o2)

Значение null

- Переменные ссылочных типов могут принимать специальное значение null
- Пример: String s = null;
- Оно означает пустую ссылку, то есть адрес, который никуда не указывает
- Если вызвать функцию для переменной, которая имеет значение null, то произойдет ошибка NullPointerException

Для чего полезен null?

- Значение null может быть полезно, если мы хотим показать, что функция отработала, но получить результат не удалось
- Например, мы написали функцию, которая ищет строку нужной длины среди некоторого набора строк
- Но такой строки не оказалось
- В этом случае функция может вернуть null, а вызывающий код проверить, что результат равен null и, например, напечатать сообщение, что ничего не найдено

Для чего полезен null?

```
public static String findString(int length) {
    // код, который делает return, если нашел строку

    // в конце делается return null если
    // ничего не найдено
    return null;
}

public static void main(String[] args) {
    if (findString(4) == null) {
        System.out.println("Ничего не найдено");
    }
}
```

Символьный тип char

- Кроме строкового типа, в Java есть символьный тип char
- Это примитивный тип
- Размер переменной – 2 байта
- Его переменные могут хранить один символ
- Литералы заключены в одинарные кавычки: ‘a’, ‘5’, ‘\\’, ‘\n’
- char lineSeparator = ‘\n’;

Символьный тип char

- У строк можно брать символ по порядковому номеру (отсчитывается от 0)
- String s = “ABCDE”;
- char secondSymbol = s.charAt(1); // B
- char lastSymbol = s.charAt(5);
  // ошибка при исполнении программы –
  // выход за границы строки
- Правильно:
  char lastSymbol = s.charAt(s.length() - 1);

Функции для работы с символами

- Стандартный класс Character:
  - boolean isDigit(char c) – проверка, что цифра
  - boolean isLetter(char c) – проверка, что буква
  - boolean isLetterOrDigit(char c) – что буква или цифра
  - boolean isLowerCase(char c) – что буква в нижнем регистре
  - boolean isUpperCase(char c) – что буква в верхнем регистре
- Пример:

```
boolean isDigit = Character.isDigit('4');   // true
```

Пробельные символы

- boolean Character.isWhitespace(char c) – проверка, что пробельный символ
- **Пробельными символами** считаются пробел, табуляция и перевод строки

Функции работы с символами

- Стандартный класс Character:
  - char toUpperCase(char c) – перевод в верхний регистр
  - char toLowerCase(char c) – перевод в нижний регистр

- Если символ уже в этом регистре, или не буква, то выдается сам символ
- **Пример:**
- ```
  char lowerCaseChar1 = Character.toLowerCase('A');   // a
  char lowerCaseChar2 = Character.toLowerCase('a');   // a
  char upperCaseChar1 = Character.toUpperCase('A');   // A
  char upperCaseChar2 = Character.toUpperCase('a');   // A
  ```
  ```

Пример работы со строками

```
Scanner scanner = new Scanner(System.in);
String name = scanner.nextLine();
if (Character.isLowerCase(name.charAt(0))) {
    System.out.println(
      “Имя должно начинаться с заглавной буквы!”);
    return;
}
```

Проход по всем символам строки

- ```
  Scanner scanner = new Scanner(System.in);
  String s = scanner.nextLine();

  for (int i = 0; i < s.length(); ++i) {
      char c = s.charAt(i);
      // работаем с текущим символом c
  }
  ```
  ```

Задача «Подсчет символов»

- Прочитать с консоли строку
- Вывести число букв в этой строке
- Вывести число цифр в этой строке
- Вывести число пробелов в этой строке
- Вывести число остальных символов в строке

Задача на курс «Макс. подстрока»

- Написать функцию, которая ищет в строке подстроку максимальной длины, состоящую из одного и того же символа, и выдает эту максимальную длину
- Например, есть строка "аааббдеггггв", должно выдаться число 4, потому что есть 4 подряд символа «г», и это максимальная подстрока, где подряд идет один и тот же символ
- Функция должна работать без учета регистра

Задача на курс «Палиндром»

- Объявить некоторую строковую переменную в программе
- Проверить, что данная строка является палиндромом – то есть читается одинаково слева направо и справа налево.
- При проверке не учитывать регистр символов, учитывать только буквы
- Пример палиндрома: «Аргентина манит негра»

- **Требование**: сделать без создания новой строки и без удаления символов из строки