## День 19

Темы:

[HashMap - Введение](https://youtu.be/qlr3fn9FGnk)

[HashSet - Введение](https://youtu.be/dIIVc-yGcq4)

[HashMap - Как устроен?](https://youtu.be/BbcMKFXprP8)

В уроке “HashMap - Как устроен?” разбирается внутренняя работа класса HashMap и то, как с помощью хэширования эта структура данных позволяет нам мгновенно проверять наличие объекта.

Отдельного видео по тому, как работает класс HashSet нет, потому что в Java реализация класса HashSet использует в себе класс HashMap. Простыми словами, HashSet в Java - это HashMap, у которого в качестве значений всегда null. Поэтому, все, что мы разобрали про работу HashMap полностью применимо к работе HashSet.

**Задачи:**

1. В папке resources находится файл dushi.txt. Этот файл содержит в себе полный текст произведения “Мертвые души” Н. В. Гоголя. Вам необходимо прочитать этот файл и, используя структуру данных HashMap, подсчитать, сколько раз в этом произведении было использовано каждое слово. После того, как подсчет будет произведен, вам необходимо вывести в консоль 100 самых часто используемых слов. Изучите вывод и посмотрите, сколько раз в произведении было использовано слово “Чичиков” (фамилия главного героя). Это число выпишите в коде в виде комментария (например: // Чичиков - 120).

Для того, чтобы класс Scanner считывал слова, разделяя текст по знакам препинания, используйте следующий код:

Scanner scanner = **new** Scanner(text); // в text находится файл dushi.txt

scanner.useDelimiter("[.,:;()?!\"\\s–]+");

Метод useDelimiter() позволяет задать разделитель, который используется классом Scanner. По умолчанию, класс Scanner использует пробел в качестве разделителя.

**Поэтому, такой код:**

String text = "hello goodbye 123";

Scanner scanner = **new** Scanner(text);

**while** (scanner.hasNext())

System.out.println(scanner.next());

**Выдаст такой результат:**

hello

goodbye

123

Если слова в нашем тексте будут разделяться тремя дефисами (---), например, мы сможем задать этот новый разделитель для Scanner.

String text = "hello---goodbye---123";

Scanner scanner = **new** Scanner(text);

scanner.useDelimiter("---");

**while** (scanner.hasNext())

System.out.println(scanner.next());

**Результат в консоли будет аналогичным (благодаря useDelimiter(---)):**

hello

goodbye

123

В тексте Н. В. Гоголя большое количество различных знаков препинания, поэтому, чтобы выделить только слова из этого текста, нам недостаточно указать какой-то один разделитель в useDelimiter(). Вместо этого, в useDelimiter() мы указываем так называемое регулярное выражение (англ. regular expression). Регулярные выражения - это особый язык, который описывает паттерны в тексте.

Наше регулярное выражение ("[.,:;()?!\"\\s–]+") описывает любую комбинацию знаков препинания и пробелов, которые могут встретиться в тексте между словами. Поэтому, такой разделитель позволяет нам эффективно выделять слова из текста. Регулярные выражения - большая тема и требует отдельного изучения (о них я говорю в курсе “[Продвинутая Java](https://www.udemy.com/course/javarussia/?referralCode=A36A8387475CAFDC6776)”). Для этого задания, нам не надо самостоятельно составлять регулярное выражение. Достаточно лишь применить данное регулярное выражение.

2. В папке resources находится файл taxi\_cars.txt. Этот файл содержит в себе данные о местоположении 500 машин такси (каждая машина такси находится на новой строке).

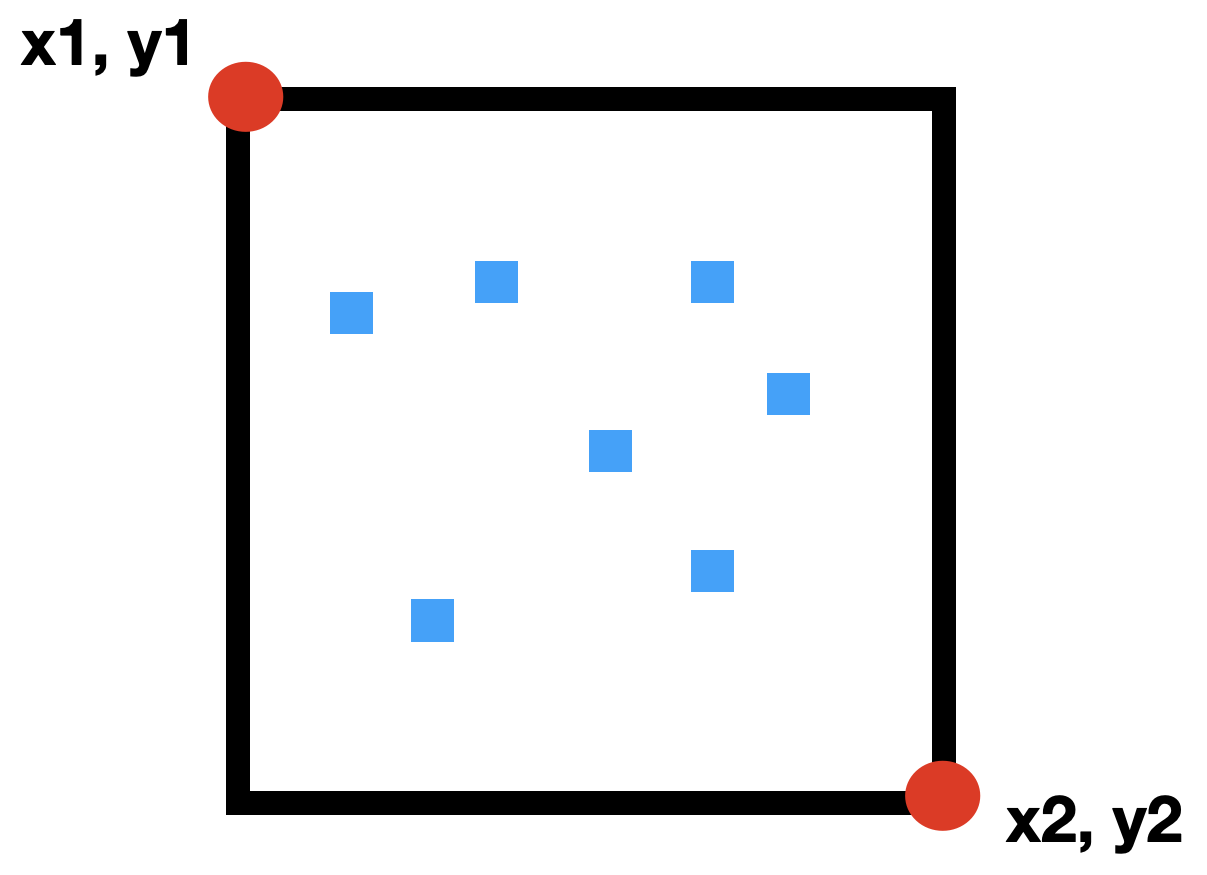
**Формат данных в файле следующий:**

id\_машины координата\_по\_x координата\_по\_y

Каждая строка в файле содержит 3 целых числа, разделенных пробелами — уникальный идентификатор машины такси, координата машины по оси X и координата машины такси по оси Y. Значения каждой из координат находятся в диапазоне от 0 до 99.

Для вашего удобства, в пакете task2 реализован класс - Point. Класс Point необходим для представления точек с координатами x,y в нашей программе. Местоположение машины такси — объект класса Point.

Вам необходимо считать строки из файла taxi\_cars.txt в HashMap<Integer, Point>. В качестве ключей используются id машин из файла, а в качестве значений используется объект Point, созданный для двух координат текущей машины. После того, как все машины будут лежать в объекте HashMap, вам необходимо запросить от пользователя 4 числа - координаты квадрата (x1, y1, x2, y2). Первые два числа — координаты первой точки квадрата, вторые два числа — координаты второй точки квадрата (см. картинку ниже).



На этой картинке видно, что квадрат задается двумя точками, которые вводятся пользователем с клавиатуры. Синими квадратиками изображены машины такси, которые оказались внутри квадрата.

Для квадрата, введенного пользователем, вам необходимо, используя сформированный объект HashMap<Integer, Point>, определить, какие машины попали внутрь этого квадрата. Вам необходимо вывести в консоль идентификаторы машин, которые находятся внутри квадрата и общее количество машин в квадрате. Машины, лежащие на грани квадрата, не учитываются.

Алгоритм должен работать для любого корректно введенного квадрата.