Продолжаем паттерны. Лямбда выражения и коллекции. Stream API

Практика 35 - Айрат Хасьянов

Разминка

- Обсудим 2 фрагмента кода

```
if(!set.contains(item)) {
    set.add(item);
    // do something
} else {
    // do something else
}
```

VS

```
if(set.add(item)) {
    // do something
} else {
    // do something else
}
```

Минутка Паттерна

- Command - Альбина Галимзянова

Вывод чисел

```
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5, 6);

for (int number : numbers) {
System.out.println(number);
}
```

Это Лямбда!

```
numbers.forEach((Integer value) -> System.out.println(value));
```

Но это не предел!

```
numbers.forEach((Integer value) -> System.out.println(value));
```

```
numbers.forEach(value -> System.out.println(value));
```

```
numbers.forEach(System.out::println);
```

Посчитаем сумму?

```
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5, 6);

public int sumAll(List<Integer> numbers) {
    int total = 0;
    for (int number : numbers) {
        total += number;
    }
    return total;
}
```

А теперь только четные?

```
public int sumAllEven(List<Integer> numbers) {
    int total = 0;
    for (int number : numbers) {
        if (number % 2 == 0) {
            total += number;
        }
    }
    return total;
}
```

А теперь только ненулевые!

```
public int sumAll(List<Integer> numbers, Predicate<Integer> p) {
    int total = 0;
    for (int number : numbers) {
        if (p.test(number)) {
            total += number;
        }
    }
    return total;
}

   sumAll(numbers, n -> true);
   sumAll(numbers, n -> n % 2 == 0);
   sumAll(numbers, n -> n != 0);
```

Анонимный класс по-новому

```
interface GreetingService {
   void sayMessage(String message);
}
```

```
GreetingService gs = message ->
 System.out.println("Hello " + message);

 gs.sayMessage("Vladimir Ilyich!");
```

Задачки на коллекции

1. Напишите методы union и intersect для множеств

2. Создайте множество, где имена студентов будут отсортированы по алфавиту

3. Сравните быстродействие Вашего стека и стека из java Collections

Дополнительные задачки

1. Что делает код: myMap.put(key1, myMap.put(key2, myMap.get(key1)));
2. Как найти максимальный элемент в коллекции?
3. Используйте EnumSet для хранения значений типа Enum, почему он быстрее, чем Set<Enum>?

Секрет эффективности

- Вместо Stack пользуйтесь ArrayDeque, вместо Vector — ArrayList, вместо Hashtable — HashMap

Домашнее задание

1. Изучить тьюториал: http://www.deadcoderising.com/java-8-no-more-loops/, использовать Stream API при решении следующих задач!

2. Используйте java.util.Map для того, чтобы посчитать количество 1, 2, 3 и так до 12 в выборке из 1000000 случайных чисел.

3. Используйте свою реализацию Comparator, чтобы выстроить пассажиров в java.util.PriorityQueue с учетом серебряных, золотых и платиновых карт, а также всех мультимиллиардеров в конец очереди ;)

4. Напишите два итератора, которые позволяют обходить вашу приоритетную очередь, ка в порядке приоритета посадки, так и в алфавитном порядке для распечатки списка пассажиров. Продемонстрируйте обход коллекции, используя разные итераторы.

5. (*) Постройте генеалогическое древо из файла d формате: имя - имя мамы - имя папы\n

6. Используйте метод subList для очистки каждых 5 элементов через 5 элементов, которые остаются нетронутыми.