

## Домашнее задание

### по теме «Абстрактные классы и интерфейсы. Лямбда выражения».

#### Формулировка задания:

1. Предусмотреть функциональный интерфейс:

```
interface ByCondition {  
    boolean isOk(int number);  
}
```

В функциональном интерфейсе обязательно проставить аннотацию.

2. Создать класс Sequence для последовательности со следующим методом:

```
public static int[] filter(int[] array, ByCondition condition) {  
    ...  
}
```

Данный метод возвращает массив, который содержит элементы, удовлетворяющие логическому выражению в condition.

3. В main в качестве condition подставить:

- проверку на четность элемента
- проверку, является ли сумма цифр элемента четным числом.

4. Программа реализуется в отдельной ветке git homeworks/homework010.

При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл .gitignore.

Программа локально коммитится и публикуется в репозиторий GitHub на проверку.

#### Дополнительная задача:

1. Изучите структуру Optional.
2. Реализуйте generic-класс Pair, похожий на Optional, но содержащий пару элементов разных типов и не запрещающий элементам принимать

значение null.

3. Реализуйте методы `getFirst()`, `getSecond()`, `equals()` и `hashCode()`, а также статический фабричный метод `of()`. Конструктор должен быть закрытым (`private`).

4. С корректно реализованным классом `Pair` должен компилироваться и успешно работать следующий код:

```
Pair<Integer, String> pair = Pair.of(1, "hello");
```

```
Integer i = pair.getFirst(); // 1
```

```
String s = pair.getSecond(); // "hello"
```

```
Pair<Integer, String> pair2 = Pair.of(1, "hello");
```

```
boolean mustBeTrue = pair.equals(pair2); // true!
```

```
boolean mustAlsoBeTrue = pair.hashCode() == pair2.hashCode(); // true!
```

5. Программа реализуется в отдельной ветке `git homeworks/homework010Addition`. При сохранении состояния программы (коммиты) пишется сообщение с описанием хода работы по задаче.

В корне папки с программой должен быть файл `.gitignore`.

Планируемый результат:

1. Ссылка на программу в репозитории GitHub;
2. Отчёт со скринами выполнения задач - постановка задачи, код задачи и результат в консоли IntelliJ Idea.

Описания плана работы:

Выполнение задания в соответствии с формулировкой требований к задаче. Задание является первым шагом к объектному моделированию предметной области с последующей реализацией на Java.

Перечень инструментов, необходимых для реализации деятельности:

Персональный компьютер, JDK 17 (либо OpenJDK 17), IntelliJ Idea для разработки на Java, GIT, Tortoise GIT.