Домашнее задание:

1. Ознакомится с материалами:
   1. [Reflection and Dynamic Proxy :](http://tutorials.jenkov.com/java-reflection/index.html)
   2. [Java Rush reflection Api](https://javarush.ru/groups/posts/513-reflection-api-refleksija-temnaja-storona-java)
2. Практика:

**Задача 1:**

Имплементировать следующий интерфейс в классе CalculatorImpl

public interface Calculator{

/\*\*

\* Расчет факториала числа.

\* @param number

\*/

int calc (int number);

}

**Задача 2:**

Вывести на консоль все методы класса, включая все родительские методы

(включая приватные)

**Задача 3:**

Вывести все геттеры класса

**Задача 4:**

Проверить что все String константы имеют значение = их имени

**public static final** String ***MONDAY*** = **"MONDAY"**;

**Задача 5:**

Реализовать кэширующий прокси

**Задача 6:**

Создать свой аннотацию Metric. Реализовать proxy класс PerformanceProxy, который в случае если метод аннотирован Metric будет выводить на консоль время выполнения метода.

Т.е написав

public interface Calculator{

/\*\*

\* Расчет факториала числа.

\* @param number

\*/

@Metric

int calc (int number);

}

И использовав его

Calculator  calculator=new PerformanceProxy(new Calculator()));

System.out.println(calculator.calc(3));

Должны увидеть:

Время работы метода: ххххх (в наносек)

6

**Задача 7:**

Реализовать следующий класс по документации

public class BeanUtils {

    /\*\*

     \* Scans object "from" for all getters. If object "to"

     \* contains correspondent setter, it will invoke it

     \* to set property value for "to" which equals to the property

     \* of "from".

     \* <p/>

     \* The type in setter should be compatible to the value returned

     \* by getter (if not, no invocation performed).

     \* Compatible means that parameter type in setter should

     \* be the same or be superclass of the return type of the getter.

     \* <p/>

     \* The method takes care only about public methods.

     \*

     \* @param to   Object which properties will be set.

     \* @param from Object which properties will be used to get values.

     \*/

    public static void assign(Object to, Object from) {... }

}