Домашно Classes and Interfaces

1. Даден е интерфейсът **IEvaluatorFactory**, който създава инстанции от тип **IEvaluator**. Целта на всеки "evaluator" е да определи стойност на база зададени параметри. Параметрите се задават чрез **add()**. Стойността се изчислява, чрез **evaluate()**. Изчислението на стойността в **evaluate()** зависи от типа на IEvaluator инстанцията.

```
/**
 *
*/
public interface IEvaluator {

	/**
	* @param d
	*            adds d as a parameter for the evaluation
	*/
	void add(double d);

	/**
	* @return the evaluated value
	*/
	Double evaluate();

}
```

```
/**
 * Interface for creating different evaluators.
 */
public interface IEvaluatorFactory {

	/**
	 * @return a new evaluator that sums the given doubles.
	 *         <p>
	 *         Example:
	 *         </p>
	 *         <code>
	 * IEvaluator eval = createSumEvaluator();
	 * eval.add(3);
	 * eval.add(4);
	 * eval.add(5);
	 * eval.evaluate() must result in 3 + 4 + 5
	 *         </code>
	 */
	public IEvaluator createSumEvaluator();

	/**
	 * @return a new evaluator that brings the given doubles to the power of 2
	 *         and adds the result to the current evaluated value.
	 *         <p>
	 *         Example:
	 *         </p>
	 *         <code>
	 * IEvaluator eval = createPowerOnEvaluator();
	 * eval.add(3);
	 * eval.add(4);
	 * eval.add(5);
	 * eval.evaluate() must result in 3^2 + 4^2 + 5^2 = 50
	 * </code>
```

```
     */
    public IEvaluator createPowerOnEvaluator();

    /**
     * @param power
     * @return a new evaluator that brings the given doubles to the power of
     *         "power" and adds the result to the current evaluated value.
     *         <p>
     *         Example:
     *         </p>
     *         <code>
     * IEvaluator eval = createPowerOnEvaluator(3);
     * eval.add(3);
     * eval.add(4);
     * eval.add(5);
     * eval.evalute() must result in 3^3 + 4^3 + 5^3 = 216
     * </code>
     */
    public IEvaluator createPowerOnEvaluator(double power);

    /**
     * @return a new evaluator that gives the fibonacci number that is most
     *         closely to the sum of the added values.
     *         <p>
     *         Example:
     *         </p>
     *         <code>
     * IEvaluator eval = createFibanociEvaluator();
     * eval.add(10);
     * eval.evaluate() must result in 8
     * </code>
     *
     *         <code>
     * IEvaluator eval = createFibanociEvaluator();
     * eval.add(100);
     * eval.add(30);
     * eval.add(1);
     * eval.evaluate() must result in 144 since the added sum is 131
     * </code>
     */
    public IEvaluator createFibonaciEvaluator();
}
```

Да се създадът две имплементации на интерфейса **IEvaluatorFactory** – **AbsoluteEvaluatorFactory** и **DirectEvaluatorFactory**.

AbsoluteEvaluatorFactory трябва да създава инстанции на **IEvaluator**, които да преобразуват всяко подадено им число в положително и след това да продължат с изчислинията.

DirectEvaluatorFactory трябва да създава инстанции на **IEvaluator**, които да не преобразуват подадените числа и да работят с тях директно, независимо дали са положителни или отрицателни.