Homework 7

Задание

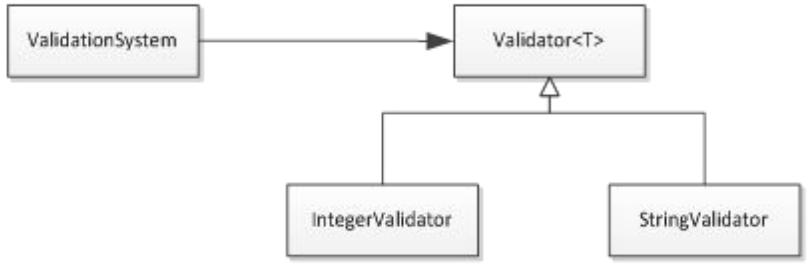
Необходимо реализовать класс валидации различных типов данных по некоторым простым правилам. Для упрощения, определим валидаторы только для двух типов Integer и String. Правила валидации такие:

- Числа целого типа должны принадлежать интервалу [1,10].
- Строка должна начинаться с заглавной буквы.

Класс получает на вход данные для валидации и в случае, если данные не соответствуют правилам, выдает исключение ValidationFailedException.

Для валидации строки использовать регулярные выражения.

Пример см Тесты.



Класс должен быть готов к расширению – добавление нового валидатора должно сводится к добавлению нового класса-валидатора и регистрации его в ValidationSystem.

Используйте Generics.

Тесты

```
@Test
public void testValidateInt () throws ValidationFailedException {
  ValidationSystem.validate(1);
  ValidationSystem.validate(5);
  ValidationSystem.validate(10);
}

@Test (expected = ValidationFailedException.class)
public void testValidateIntFail s() throws ValidationFailedException {
  ValidationSystem.validate(11);
}

@Test (expected = ValidationFailedException.class)
public void testValidateIntFails2 () throws ValidationFailedException {
  ValidationSystem.validate(0);
}

@Test
public void testValidateString () throws ValidationFailedException {
  ValidationSystem.validate(“Hello");
  ValidationSystem.validate(“Hello world, abc");
}

@Test (expected = ValidationFailedException.class)
public void testValidateStringFails() throws ValidationFailedException {
  ValidationSystem.validate(“hello");
}

@Test (expected = ValidationFailedException.class)
public void testValidateStringFails2() throws ValidationFailedException {
  ValidationSystem.validate(“”);
}
```