**Задача 1**

Создать функциональный интерфейс Filter с методом filter. Метод должен принимать в качестве параметра значение Integer и возвращать логическую переменную.

Создать класс FilterHelper, содержащий статический метод, принимающий в качестве параметра список типа Integer и параметр типа Filter. Метод должен фильтровать входящий список типа Integer и возвращать отфильтрованный список на основе параметра-метода Filter.

Реализовать тестирование созданного статического метода, передавая в него различные входные списки и лямбда-выражения, описывающие реализацию созданного фильтра на основе таблицы ниже:

|  |  |  |
| --- | --- | --- |
| List<Integer>, передаваемый в параметрах | Функциональная логика переменной Filter | Ожидаемый List<Integer> в результате |
| -20, 0, 4, 5, 7, 9 | фильтрует, оставляя значения, которые меньше 5 | -20, 0, 4 |
| -20, 0, 4, 5, 7, 9, 10, 15, 25 | фильтрует, оставляя значение, которые больше 10 | 15, 25 |
| -20, 0, 4, 5, 7, 9, 10, 15, 25 | фильтрует, оставляя значения, содержащиеся в списке (-20, 4, 9, 12, 15) | -20, 4, 9, 15 |
| -10, -7, 5, 15, 5, -2, 4, 15 | фильтрует, оставляя только значения, которые в списке не повторяются. | -10, -7, -2, 4 |

**Задача 2**

Создать функциональный интерфейс Modifier, с методом modify. Метод должен принимать в себя обобщенный тип T, проводить операции над ним и возвращать результат операции типа T.

Создать класс Modifiers. В нем создать 3 публичных статических поля, которые будут содержать лямбда-выражения и реализовывать анонимный метод интерфейса Modifier согласно таблице.

|  |  |  |
| --- | --- | --- |
| Название поля | Оперируемый тип | Операция |
| invert | Boolean | Инвертирование переменной |
| absolute | Integer | Абсолютное значение переменной (модуль) |
| reverse | String | Возврат строки в обратном порядке |

Реализовать тестирование данных полей-методов, создав один класс тестов с тремя тестами, каждый из которых тестирует в отдельном методе свой оперируемый тип.

**Задача 3**

Создать функциональный интерфейс Func с методом execute. Метод должен принимать в себя обобщенный тип K, проводить операции согласно реализации метода и возвращать результат операции типа V. Типы K и V могут быть разными.

Создать класс Functions. В нем создать 4 публичных статических поля, которые будут реализовывать функции согласно таблице. Типы дженериков K и V необходимо выбирать максимально возможно общими.

|  |  |
| --- | --- |
| Название поля | Функция |
| toStringFunction | Функция должна для любого принимаемого объекта возвращать результат выполнения метода toString() у этого объекта |
| summFunction | Функция должна для передаваемого списка целых чисел вычислять их сумму и возвращать ее. |
| onlyEqualsObjects | Функция должна для передаваемого списка объектов определять, являются ли все объекты эквивалентными. Возвращать true, если да, или false в обратном случае. Для пустого списка функция должна возвращать true |
| maxObject | Функция должна для передаваемого списка **сравниваемых** объектов вычислять максимальный объект и возвращать его в качестве результата |

Реализовать тестирование написанных полей, являющихся анонимными методами.

Для toStringFunction протестировать поле, используя классы:

* Object (для получения ожидаемого результата посмотреть реализацию toString() в Object)
* String
* Создать локальный класс Cat с полями String name, int age, метод toString() переопределить, чтобы он выводил информацию в формате “Имя: %s, Возраст: %d”

Для summFunction протестировать поле, используя входные данные:

* Список целых чисел [1, 2, 3, 4, 5, 10, 9, 8, 7, 6], ожидаемый результат: 55

Для onlyEqualsObjects протестировать поле, используя классы:

* Object (создать несколько одинаковых объектов и один отличающийся, проверить на различных комбинация)
* Проверить на пустом списке
* String (проверить на трех одинаковых строках, на двух одинаковых и одной отличающейся)
* Создать локальный класс Cat с полями String name, int age, переопределить equals() и hashCode(). Проверить на нескольких одинаковых/разных объектах локального класса

Для maxObject протестировать поле, используя классы:

* Double (проверить, что возвращается максимальный элемент)
* String (проверить, что возвращается строка, являющаяся по алфавиту последней в списке)