## Лабораторная работа №1

## *ЛР №1 выдается студентам в одном варианте. Срок выполнения - 2 недели. Форма представления отчета - программа, написанная на языке Java (код и скрины примера работы программы).*

**Задание №1:**

Реализовать аналоги стандартных интерфейсов для работы с лямбда-функциями:

* Абстрактные классы (или интерфейсы)
  + Function1 — функция одного аргумента (“f(x)”)
  + Function2 — функция от двух аргументов (“f(x, y)”)
  + Predicate — предикат для одного аргумента
  + Нужна какая-нибудь иерархия, связывающая эти классы (интерфейсы)
  + Сигнатуры всех классов и методов должны быть максимально общими
* Должны быть реализованы методы:
  + Function1.compose — композиция — принимает “Function1 g”, возвращает “g(f(x))”, например, int x = f.compose(g).apply(239);
  + Function2.compose — композиция — принимает “Function1 g”, возвращает “g(f(x, y))”
  + Function2.bind1 — bind первого аргумента — принимает первый аргумент, возвращает “f(\_, y)”
  + Function2.bind2 — bind второго аргумента — принимает второй аргумент, возвращает “f(x, \_)”
  + Function2.curry — каррирование, конвертация в “Function1” — например, f = (x, y) -> x + y, f(5) = x -> x + 5
* Предикаты:
  + Predicate.or и Predicate.and
  + принимают один предикат в качестве аргумента, возвращают предикат, который ведет себя, как дизъюнкция/конъюнкция текущего предиката и предиката-аргумента
  + семантика ленивая, как у || и &&
  + Predicate.not принимает 0 аргументов, возвращает предикат-отрицание текущего предиката
  + Константные предикаты:

Predicate.ALWAYS\_TRUE

Predicate.ALWAYS\_FALSE

* + Можно не поля, а методы
* Отдельный класс Collections с методами:
  + map — принимает f и a, применяет f к каждому элементу ai и возвращает список [f(a1), ..., f(an)]
  + filter — принимает p и a, возвращает список, содержащий элементы ai, на которых p(ai) == true
  + takeWhile — принимает p и a, возвращает список с началом a до первого элемента ai, для которого p(ai) == false
  + takeUnless — то же, что и takeWhile, только для p(ai) == true
  + foldr / foldl — принимает функцию двух аргументов, начальное значение и коллекцию
  + a - Iterable

## Критерии оценивания и шкала оценки

|  |  |
| --- | --- |
| **Оценка** | **Критерии выставления оценки** |
| «Отлично»  (8-10) | Программа решает поставленную задачу, имеет корректное оформление согласно кодовым соглашениям и содержит не более 2 архитектурных, смысловых или программных ошибок. |
| «Хорошо»  (6-7) | Программа решает поставленную задачу, имеет корректное оформление согласно кодовым соглашениям, имеет 3-4 архитектурных, смысловых или программных ошибки |
| «Удовлетворительно»  (4-5) | Программа решает поставленную задачу, имеет корректное оформление согласно кодовым соглашениям, имеет 5 архитектурных, смысловых или программных ошибок |
| «Неудовлетворительно» (0-3) | Программа не решает поставленную задачу и/или не имеет корректного оформления согласно кодовым соглашениями и/или содержит более 5 архитектурных, смысловых или программных ошибок. |