Лабораторная работа №4
«Неявные преобразования типов в Scala»

Скоробогатов С.Ю.

16 апреля 2016 г.

1 Цель работы

Целью данной работы является приобретение навыков использования неявных преобразований типов для разработки на языке Scala обобщённых классов, набор операций которых определяется их фактическими типовыми параметрами.

2 Исходные данные

...
`http://dcsobral.blogspot.ru/2010/06/implicit-tricks-type-class-pattern.html`

3 Задание

...

Таблица 1: Варианты классов

1	Класс `SuperNumber[T]`, представляющий число произвольной разрядности в системе счисления, цифрами которой выступают значения типа `T`. Если `T` – целочисленный тип Scala или `Bool`, то для `SuperNumber[T]` должна быть доступна операция сложения. Если `T` – `Bool`, то дополнительно должны работать операции поразрядного И и ИЛИ.
2	Класс `Fib[T]`, представляющий i-тый член последовательности Фибоначчи типа `T`. Конструктор класса `Fib[T]` принимает два первых члена последовательности и создаёт объект, соответствующий первому члену. Метод `next` возвращает новый объект, соответствующий следующему члену. Если тип `T` – целочисленный, то очередной член последовательности вычисляется как сумма двух предыдущих. Если тип `T` – строка, то вместо сложения используется конкатенация.
3	Класс `Vector[T <: Product]`, представляющий вектор в двух или трёхмерном пространстве. Если тип `T` представляет пару или тройку, элементы которой имеют одинаковый числовой тип, то для `Vector[T <: Product]` должны быть доступны операции сложения и скалярного умножения. Дополнительно, для троек должна быть реализована операция векторного умножения.
4	Класс `SuperStack[T]`, представляющий неизменяемый стек с операциями `push`, `pop` и `empty`, реализованный через список. В случае, если `T` – числовой тип, для стека должны быть также доступны операции `min` и `max`, работающие за константное время.
5	Класс `Bijection[T]`, представляющий неизменяемое взаимнооднозначное отображение заданного конечного подмножества значений типа `T` в себя с операциями применения отображения к значению (возвращает `Option[Bijection[T]]`) и обмена значений, в которые отображаются два элемента (`swap`). Конструктор класса должен получать в качестве параметра множество и порождать тождественное отображение. Для `Bijection[String]` должен быть доступен метод переворачивания всех строк в отображении, работающий за константное время. Для `Bijection[Int]` должен быть доступен метод прибавления заданного числа к каждому элементу множества, на котором определено отображение (он тоже должен работать за константное время).
6	Класс `MegaQueue[T]`, представляющий неизменяемую очередь с операциями `enqueue`, `dequeue` и `empty`, реализованную через два списка. В случае, если `T` – числовой тип, для очереди должна быть также доступна операция `max`, работающая за константное время.
7	Класс `QuadraticEquation[T]`, представляющий квадратное уравнение с коэффициентами типа `T` и оперцией `solve`, возвращающей список найденных корней типа `T`. Операция `solve` должна быть доступна в случае, если `T` – числовой тип из стандартной библиотеки языка Scala. Кроме того, эта операция должна быть реализована для класса комплексных чисел, написанного самостоятельно или найденного в Интернете.

Таблица 2: Варианты классов

8	Класс `EquationSystem[T]`, представляющий систему из двух линейных уравнений от двух переменных с коэффициентами типа `T` и оперцией `solve`, возвращающей решение уравнения в виде `Option[(S, S)]`, где тип `S` зависит от типа `T`. Эта зависимость задаётся следующими правилами: если `T` – тип с плавающей точкой, то `S` совпадает с `T`; если `T` – целочисленный тип, то `S` – дробь, числитель и знаменатель которой имеют тип `T`; в остальных случаях операция `solve` недоступна. Для представления дробей следует создать отдельный класс.
9	Класс `Polynom[T]`, представляющий полином с коэффициентами типа `T` и операцией, возвращающей степень полинома. В случае, если тип `T` – числовой, для `Polynom[T]` также должны быть доступны операции дифференцирования и вычисления значения полинома в точке x.
10	Класс `MegaStack[T]`, представляющий неизменяемый стек с операциями `push`, `pop` и `empty`, реализованный через список. В случае, если `T` – числовой тип, для стека должна быть также доступна операция `average`, возвращающая среднее арифметическое элементов стека и работающая за константное время.
11	Класс `CompressedStack[T]`, представляющий неизменяемый стек с операциями `push`, `pop` и `empty`, реализованный через массив байтов. Тип `T` должен быть либо целочисленным типом, либо типом `Bool`. Конструктор стека должен принимать диапазон значений, которому должны принадлежать элементы стека. Для хранения каждого элемента нужно использовать минимально возможное количество бит, зависящее от величины диапазона.
12	Класс `CompressedQueue[T]`, представляющий неизменяемую очередь с операциями `enqueue`, `dequeue` и `empty`, реализованную через кольцевой список, который представлен массивом байтов. Тип `T` должен быть либо целочисленным типом, либо типом `Bool`. Конструктор очереди должен принимать диапазон значений, которому должны принадлежать её элементы. Для хранения каждого элемента нужно использовать минимально возможное количество бит, зависящее от величины диапазона.
13	Класс `CompressedPolynom[T]`, представляющий неизменяемый полином с коэффициентами типа `T`, хранение который реализовано через массив байтов. Тип `T` должен быть либо целочисленным типом, либо типом `Bool`. Конструктор полинома должен принимать диапазон значений, которому должны принадлежать его коэффициенты. Для хранения каждого коэффициента нужно использовать минимально возможное количество бит, зависящее от величины диапазона. В классе `CompressedPolynom[T]` должна быть реализована операция вычисления значения полинома в точке x.
14	Класс `KadaneStack[T]`, представляющий неизменяемый стек с операциями `push`, `pop` и `empty`, реализованный через список. В случае, если `T` – числовой тип, для стека должна быть также доступна операция `maxSum`, возвращающая максимальную сумму подряд идущих элементов стека и работающая за константное время.

Таблица 3: Варианты классов

15	Класс `Permutation[T]`, представляющий неизменяемую перестановку подмножества значений типа `T` с двумя операциями: получение i-го элемента и транспозиция. В случае, если `T` – строковый тип, для `Permutation[T]` доступна дополнительная операция `superStringSize`, возвращающая длину минимальной суперстроки, в которую строки, содержащиеся в перестановке, входят в том порядке, в котором они расположены в перестановке.