**Федеральное государственное образовательное бюджетное учреждение высшего профессионального образования**

**Московский технический университет связи и информатики**

**КАФЕДРА ИНФОРМАТИКИ**

**Лабораторная работа по теме 2  
Классы типов и функции на языке Haskell**

**Выполнил:**

**Студент группы БВТ1701**

**Шабанов Борис**

**Вариант 22**

**Москва 2019**

**Цель работы:** изучить класс типов и относящееся к нему набор функций, экземпляры класса типов, а также функции и их действия и влияния на типы.

**1 Классы типов**

**1.1 Класс Eq**

Класс Eq используется для типов, которые поддерживают проверку равенства. Типы, являющиеся его экземплярами, должны реализовывать функции == и /=

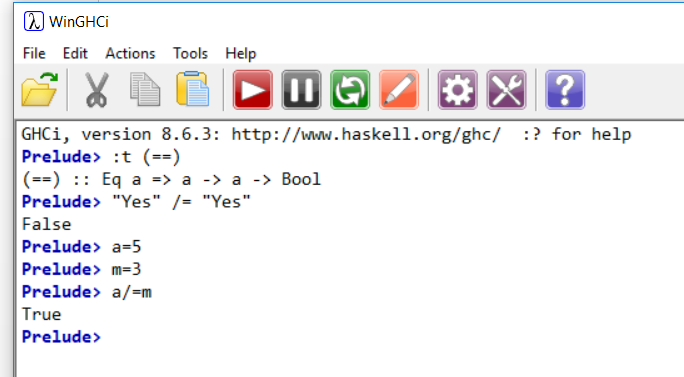


Рисунок 1 - Класс Eq.

**1.2 Класс Ord**

Все типы, упоминавшиеся ранее, за исключением функций, имеют экземпляры класса Ord. Класс Ord содержит все стандартные функции сравнения, такие как >, = и <=

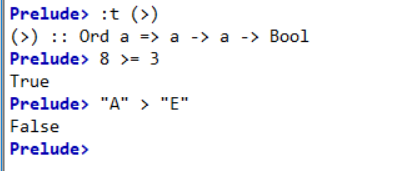


Рисунок 2 - Класс Ord .

**1.3 Класс Show**

Значения, типы которых являются экземплярами класса типов Show, могут быть представлены как строки. Она берёт значение, для типа которого определён экземпляр класса Show, и представляет его в виде строки.

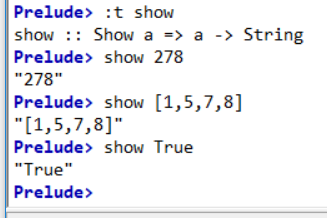


Рисунок 3 - Класс Show.

**1.4 Класс Read**

Класс Read – это нечто противоположное классу типов Show. Функция read принимает строку и возвращает значение, тип которого является экземпляром класса Read.



Рисунок 4 - Класс Read.

**1.5 Класс Enum**

Экземплярами класса Enum являются последовательно упорядоченные типы; их значения можно перенумеровать. Основное преимущество класса типов Enum в том, что мы можем использовать его типы в интервалах списков. Кроме того, у них есть предыдущие и последующие элементы, которые можно получить с помощью функ ций succ и pred. Типы, входящие в этот класс: (), Bool, Char, Ordering, Int, Integer, Float и Double.

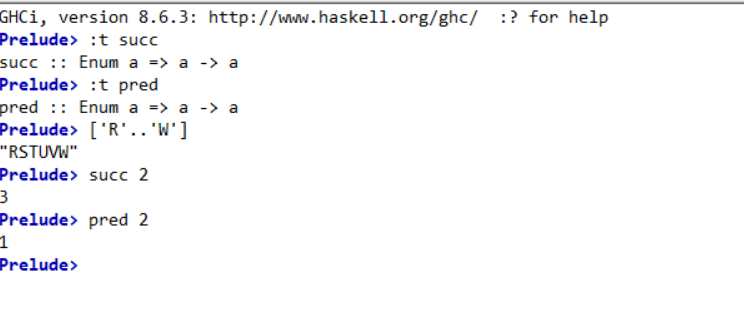


Рисунок 5 - Класс Enum.

**1.6 Класс Bounded**

Экземпляры класса типов Bounded имеют верхнюю и нижнюю границу. Функции minBound и maxBound интересны тем, что имеют тип (Bounded a) => a. В этом смысле они являются полиморфными кон стантами.

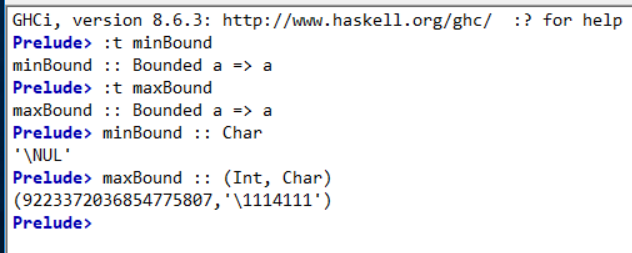


Рисунок 6 - Класс Bounded.

**1.7 Класс Num**

Класс Num – это класс типов для чисел. Его экземпляры могут вести себя как числа. Если проверить тип оператора \*, можно увидеть, что он принимает любые числа.

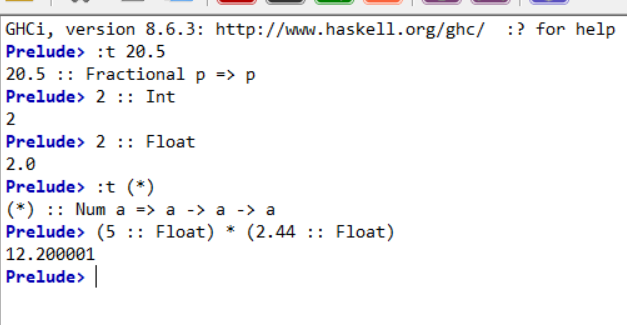


Рисунок 7 - Класс Num.

**1.8 Класс Floating**

Класс Floating включает в себя только числа с плавающей точкой, то есть типы Float и Double. Некоторые примеры: функции sin, cos и sqrt

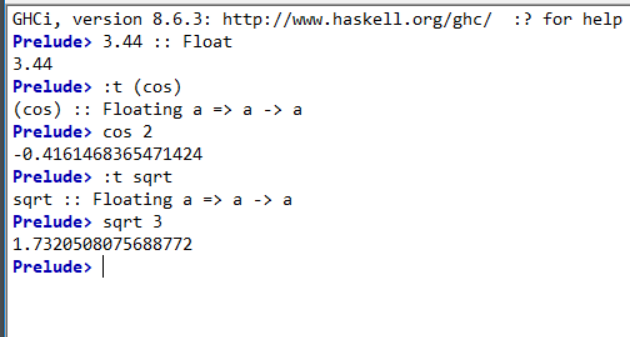


Рисунок 8 - Класс Floating.

**1.9 Класс Integral**

Класс Integral – тоже числовой класс типов. Если класс Num включает в себя все типы, в том числе действительные и целые числа, то в класс Integral входят только целые числа.

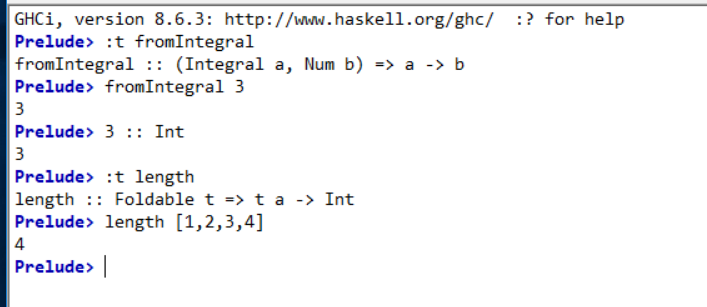


Рисунок 9 - Класс Integral.

**2 Классы Функций**

**2.1 Класс replicate**

Функция replicate берёт целое число (типа Int) и некоторый элемент и возвращает список, который содержит несколько повторений заданного элемента.

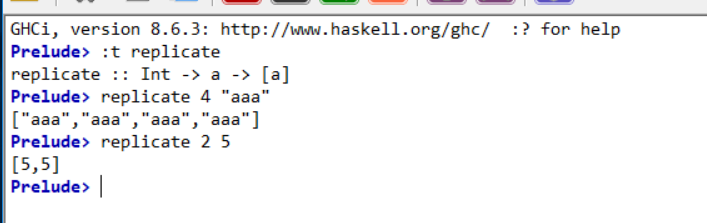


Рисунок 10 - Класс replicate.

**2.2 Класс take**

Эта функция берёт определённое количество первых элементов из заданного списка. Например, take 3 [5,4,3,2,1] вернёт список [5,4,3].

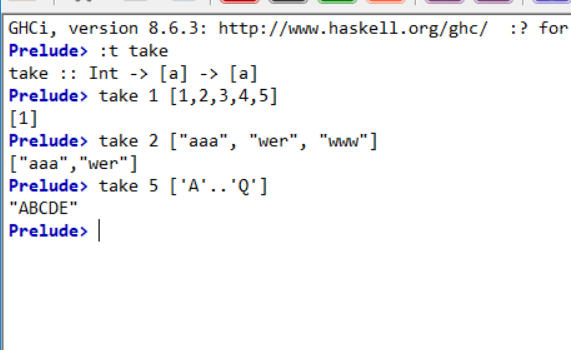


Рисунок 11 - Класс take.

**2.3 Класс reverse**

Функция reverse обращает список, выстраивая элементы в обратном порядке.

****

Рисунок 12 - Класс reverse.

**2.4 Класс repeat**

Функция repeat принимает на вход некоторый элемент и возвращает бесконечный список, содержащий этот элемент. Функция repeat наглядно показывает, что рекурсия может вообще не иметь базового случая, если она создаёт бесконечные списки – нам нужно только вовремя их где-нибудь обрезать.

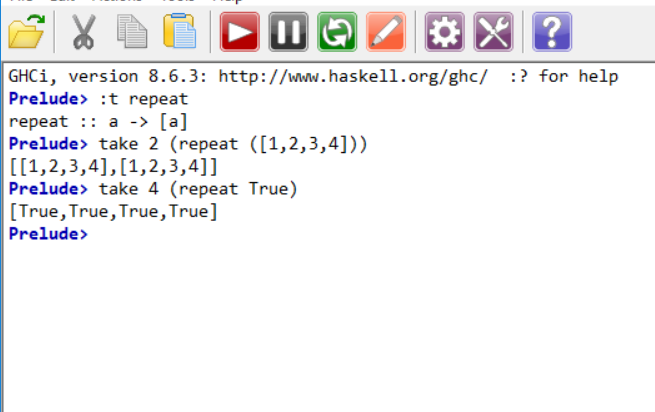


Рисунок 13 - Класс repeat.

**2.5 Класс zip**

Функция zip берёт два списка и стыкует их, образуя список пар (по аналогии с тем, как застёгивается замок-молния). Так, например, zip [1,2,3] ['a','b'] вернёт список [(1,'a'),(2,'b')].

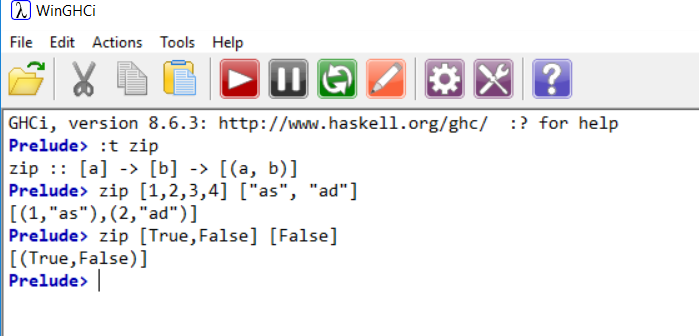


Рисунок 14 - Класс zip.

**2.6 Класс elem**

Давайте реализуем ещё одну функцию из стандартной библиотеки – elem. Она принимает элемент и список и проверяет, есть ли заданный элемент в этом списке

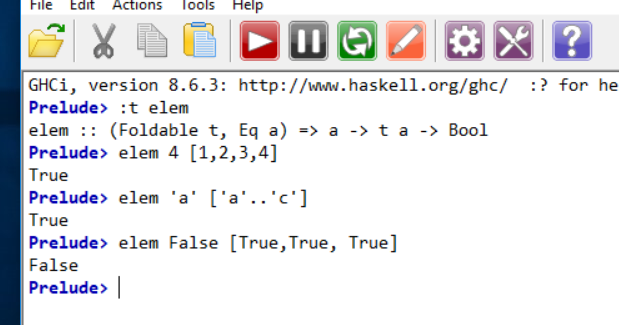


Рисунок 15 - Класс elem.