

ФИТ НГУ, курс Haskell, осенний семестр 2019

Хэш-таблица

Общие сведения

Хэш-таблица - это ассоциативный контейнер, позволяющий хранить пары ключ-значение, при этом для ключей должна иметь смысл операция вычисления хэша. Хэш-таблицы примечательны высокой скоростью вставки, поиска и удаления - все три операции выполняются с алгоритмической сложностью амортизированное $O(1)$.

Подробнее о хэш-таблицах: https://en.wikipedia.org/wiki/Hash_table.

Задача

1. Реализовать хэш-таблицу с заданным интерфейсом (см. раздел "Реализация"). Используйте алгоритм цепочек для решения коллизий.
2. В качестве ключа должен быть выступать любой *подходящий* Show тип (то есть достаточно написать одну функцию вычисления хэша для строк)
3. Обязательным является требование рехеширования при заполненности таблицы выше, чем ее load factor
4. Вместо вопросов необходимо самим подставить нужный тип.
5. При сдаче нужно будет продемонстрировать работу следующим образом:
 - 1) Считать из какого-то большого файла данные в формате: *ключ значение*
 - 2) Запихать их в вашу таблицу и показать несколько геттеров\сеттеров

Реализация

```
-- Возвращает пустую хэштаблицу
defaultHashTable :: Hashtable k v

-- конструирует хэштаблицу из списка пар,
удовлетворяющим всем условиям.
-- Это единственные доступные для
пользователя конструкторы. Все остальные
```

вариации получаются путем
добавления\удаления элементов из уже
существующих таблиц

```
fromList::(Show k, ??? k)=> [(k,v)]->HashTable k v
```

```
// Очищает контейнер.
```

```
clear::HashTable k v -> HashTable k v;
```

```
// Удаляет элемент по заданному ключу.
```

```
Erase::(Show k, ??? k)=>HashTable k v->k->HashTable k v
```

```
// Вставка в контейнер.
```

```
Insert::(Show k, ??? k)=>HashTable k v ->k->v->HashTable k v
```

```
// Проверка наличия значения по заданному  
ключу.
```

```
Contains::(Show k, ??? k)=>HashTable k v -> k -> Bool;
```

```
// Возвращает значение по ключу.
```

```
at::(Show k, ??? k)=>HashTable k v -> k ->???
```

```
size::(Show k, ??? k)=>HashTable k v -> Integer;
```

```
empty::(Show k, ??? k)=>HashTable k v -> Bool;
```