Лабораторная работа №2

по курсу «Языки программирования и методы программирования» (информатика, 3 семестр)

Техническое задание

Используемые термины и сокращения

АТД	Абстрактный тип данных

1. Постановка задачи

На языке C++ реализовать АТД “Словарь” на основе хеш-таблицы.Реализовать алфавитный указатель.Реализовать тип данных “Разреженный массив”.

2. Функциональные требования

2.1 АТД “Словарь" должен позволять хранить элементы любых типов. Для этого класс должен быть шаблонным.

```
template<class K, class V> class IDictionary {
public:
  ....
}
```

2.2 АТД "Словарь" должен обладать, по крайней мере, следующими методами:

Название	**Сигнатура**	**Назначение**
add	*void add(K,V)*	Добавить новый элемент
remove	*void remove(K)*	Удалить элемент
get	*V get(K)*	Получить элемент по ключу (значение)
at	*V &at(K)*	Получить элемент по ключу(ссылка)
operator[]	*V &operator[](K)*	Получить элемент по ключу(ссылка)
find	*bool find(K)*	Возвращает true, если элемент существует, false в противном случае

operator <<	*std::ostream& operator<<(std::ostream& , IDictionary& dict)*	Вывод словаря в поток
operator ==	*bool operator==(IDictionary<K,V > &)*	Сравнение словарей

2.2 Реализовать в хеш-таблице итератор, который должен обладать, по крайней мере, следующими методами:

Название	**Сигнатура**	**Назначение**
next	*void next()*	Получить следующий элемент. Если он отсутствует, вызывать исключение "end of collection".
hasNext	*bool hasNext()*	Возвращает true, если следующий элемент коллекции существует, false в противном случае

2.3 АТД "Разреженный вектор" должен позволять хранить элементы любых типов. Для этого класс должен быть шаблонным.

```
template<class T> class SparseVector{
public:
  ....
}
```

2.4 Тип данных "Разреженный вектор" должен обладать, по крайней мере, следующими методами:

Название	**Сигнатура**	**Назначение**
get	*T &get(int)*	Получить элемент по индексу
operator[]	*T operator[](int)*	Получить элемент по индексу

getRef	*T &SparseArray<T>::getRef(int key)*	Получить ссылку на элемент по индексу
set	*void set (int,T)*	Установить элемент на соответствующую позицию
append	*void append(T)*	Добавить элемент в конец
getSize	*int getSize()*	Получить размер массива
operator <<	*std::ostream &operator<<(ostream &, SparseArray<T> &arr)*	Вывод массива в поток

3. Требования к структурам данных и алгоритмам

3.1 АТД "Словарь" должен быть реализован с помощью хеш-таблицы

3.2 Для хранения данных в хеш-таблице должен быть использован DynamicArray

3.3 В хеш-таблице должно быть реализовано разрешение коллизий с помощью метода двойного хеширования.

3.4 Алгоритм "Алфавитный указатель" должен уметь работать как с размером страницы , указанным в кол-ве слов, так и с размером, указанным в кол-ве символов.

4. Требования к формату выходных данных

4.1 "Словарь" должен выводиться в формате {key:value}

4.2 "Разреженный вектор" должен выводиться в формате [elements]

5. Требования к интерфейсу:

нет

6. Требования к тестам

Основные операции должны быть покрыты тестами