Лабораторная работа №2

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

Техническое задание

Используемые термины и сокращения

|  |  |
| --- | --- |
| АТД | Абстрактный тип данных |
|  |  |
|  |  |

# Постановка задачи

На языке C++ реализовать АТД «Словарь», «Множество» на основе бинарного дерева. Решить задачи «Построение гистограммы», «Обработка разреженных векторов и матриц», «Индексирование данных». Написать unit-тесты. Реализовать пользовательский интерфейс.

# Функциональные требования

* 1. АТД должны позволять хранить элементы любых типов. Для этого классы должны быть объявлены как шаблонные.
  2. АТД «Множество» («Set») должен обладать, по крайней мере, следующими методами:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название** | **Сигнатура** | **Назначение** |
|  | Exist | bool exist(T elem) | Узнать, есть ли элемент во множестве. |
|  | Erase | void erase(T elem) | Удалить элемент из множества |
|  | Add | void add(T elem) | Добавить элемент во множество. |
|  | GetSize | int getSize() | Получить размер множества |

* 1. АТД «Словарь» («Dictionary») должен обладать, по крайней мере, следующими методами:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название** | **Сигнатура** | **Назначение** |
|  | containsKey | bool containsKey(TKey key) | Узнать, есть ли такой ключ в словаре |
|  | remove | void remove(TKey key) | Удалить элемент из словаря по ключу |
|  | Add | void add(TKey key, TElement element) | Добавить элемент с ключем в словарь. |
|  | GetCount | int getCount() | Получить количество элементов в словаре |
|  | get | TElement get(TKey key) | Получить элемент из словаря по ключу |
|  | changeElem | void changeElem(TKey key, TElement elem) | Изменить элемент в словаре по ключу |

* 1. Для решения задачи «Индексирование данных» нужно реализовать АТД «IndexDictionary». АТД должен строиться на основе ArraySequence<Person>. Индекс имеет тип <T> и должен быть относительно произвольным. АТД обладает следующими методами:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название** | **Сигнатура** | **Назначение** |
|  | contains | bool contains(T index) | Узнать, есть ли объект с таким индексом в структуре данных. |
|  | getCount | int getCount() | Получить количество проиндексированных объектов. |
|  | get | Person get(T index) | Получить объект по индексу. |

* 1. По техническим причинам структуру обрабатываемых данных необходимо зафиксировать. АТД привязан к классу “Person” со следующими методами:

string GetFirstName();

string GetMiddleName();

string GetLastName();

string GetFullName();

string GetFIO(); // Фамилия И.О.

int GetBirthYear();

int GetAge(int year); // вычислить возраст на заданный момент

* 1. Для решения задачи «Обработка разреженных векторов и матриц» необходимо реализовать АТД «Разреженная матрица». АТД должен обладать следующими методами:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Название** | **Сигнатура** | **Назначение** |
|  | set | void set(int x, int y, T value) | Изменить значение ячейки матрицы. |
|  | setToZero | void setToZero(int x, int y) | Обнулить значение ячейки матрицы |
|  | isNotZero | bool isNotZero(int x, int y) | Проверить значение ячейки на неравенство нулю |
|  | Get | T get(int x, int y) | Получить значение ячейки матрицы |
|  | getAmountOfNonZero | int getAmountOfNonZero() | Получить число ненулевых ячеек матрицы |
|  | getRows | int getRows() | Получить число строк матрицы |
|  | getColumns | int getColumns() | Получить число столбцов матрицы. |
|  | setRows | void setRows(int a) | Изменение числа строк матрцы. |
|  | setColumns | void setColumns(int a) | Изменение числа столбцов матрицы |
|  | getMatrix | Matrix<T> getMatrix() | Получение обычной матрицы. |

* 1. Для решения задачи «Построение гистограммы» необходимо реализовать класс «Гистограмма». Для создания объекта класса Histogram должны быть нужны ArraySequence<Person>, функция, возвращающаяся численный параметр для каждого члена класса Person (к примеру, Birth year) и число элементарных подмножеств, на которые будет разбита исходная последовательность Person.

# Требования к структурам данных и алгоритмам

Нет.

# Требования к интерфейсу

Нет.

# Требования к форматам входных и выходных данных

Требования к форматам входных данных

Нет.

Требования к форматам выходных данных

Нет.

# Требования к unit-тестам

* 1. Написать, по крайней мере, один тест[[1]](#footnote-1), реализующий следующий сценарий:
     1. Создать пустую последовательность (TElement = int).
     2. Проверить, что длина = 0
     3. Добавить (Append) элемент «23».
     4. Проверить, что длина = 1
     5. Проверить, что GetFisrt возвращает «23»
     6. Проверить, что GetLast возвращает «23»
     7. Проверить, что Get(0) возвращает «23», а Get(-1) и Get(1) – выбрасывают исключение.
     8. Добавить (Append) элемент «43»
     9. Проверить, что длина = 2
     10. Проверить, что GetFisrt возвращает «23»
     11. Проверить, что GetLast возвращает «43»
     12. Проверить, что Get(0) возвращает «23», Get(1) – «43», а Get(-1) и Get(2) – выбрасывают исключение.
     13. Добавить (Prepend) элемент «53»
     14. Проверить, что длина = 3
     15. Проверить, что GetFisrt возвращает «53»
     16. Проверить, что GetLast возвращает «43»
     17. Проверить, что Get(0) возвращает «53», Get(1) – «23», а Get(-1) и Get(3) – выбрасывают исключение.
     18. Проверить, что GetSubsequence(1,1) возвращает последовательность, у которой:
         1. длина =1
         2. первый и последний элемент – 23

1. [↑](#footnote-ref-1)