

Программа «FORTtest»

Программа позволяет вести учет позиций:

- «first_name» - Имя;
- «second_name» - Фамилия;
- «age» - возраст.

Учет ведется за счет небольшого набора команд:

Команда «add» - добавляет в конец списка позицию. Пример ввода в командной строке: «add ivan ivanov 24».

Команда «del» - удаляет из списка выбранную позицию. Пример ввода в командной строке: «del 4» - удаление 4 элемента в списке.

Команда «list» выводит список всех позиций. Пример ввода в командной строке: «list».

Команда «Search» - Осуществляет поиск в списке позиций. Выводит позиции, совпадающие с условиями поиска. Для удачного результата поиска поисковой запрос должен полностью совпадать с искомым элементом. Пример ввода в командной строке: «search ivan ivanov 24».

Реализация

Учет абстрактных позиций реализован с помощью шаблона класса *PositionList*, шаблону, в качестве элемента (позиции), передается пользовательский класс. *PositionList* не зависит от реализации передаваемого класса.

Для учета позиций был разработан класс *Person*, представляющий собой одну позицию из списка. Персистентность данных достигается за счет хранилища, реализованного в виде обычного txt файла. В начале выполнения программы из хранилища в вектор загружаются данные, в конце выполнения программы данные выгружаются из вектора в хранилище. Адрес хранилища указан в переменной «dataSource».

Шаблон класса «PositionList»

Поле класса (реализовано как скрытый элемент класса):

- V – список позиций (реализован через класс *vector*, входящий в STL c++);

Методы шаблона класса:

void push_back(T t). Добавляет элемент в конец списка. В качестве параметра передается абстрактный элемент.

void del_element(int i). Удаляет выбранный элемент из списка. В качестве параметра передается порядковый номер элемента.

void list(). Выводит список всех элементов.

void search(T s). Выполняет поиск элемента в списке. В качестве параметра передается абстрактный элемент, относительно которого выполняется поиск.

void read_from_storage(string s). Выполняет загрузку элементов из хранилища. В качестве параметра передается адрес хранилища, в виде строки.

void write_to_storage(string s). Выполняет выгрузку элементов в хранилище. В качестве параметра передается адрес хранилища, в виде строки.

Класс «Person»

Поля класса (реализованы как скрытые элементы класса):

- «first_name» - Имя;
- «second_name» - Фамилия;
- «age» - возраст.

Методы класса:

set_age(int a). устанавливает значение для поля «age» через параметр a;

get_age(). Возвращает значение поля «age».

set_first_name(string f). устанавливает значение для поля «first_name» через параметр f;

get_first_name(). Возвращает значение поля «first_name».

set_second_name(string s). устанавливает значение для поля «second_name» через параметр s;

get_second_name(). Возвращает значение поля «second_name».

Дружественные функции класса:

ostream& operator<<(ostream& os, Person& p). Перегрузка оператора вывода для класса Person

istream& operator>>(istream& is, Person& p). Перегрузка оператора ввода для класса Person

bool operator==(Person& p1, Person& p2). Перегрузка оператора сравнения для класса Person.

Функции программы «TEST_for_FORT»

info(). Вывод блока информации.

exec_prog(vector<Person>& P). Функция выполнения программы.