**Билет 3ы**

В C++ классы создаются с помощью ключевого слова class, а их структура делится на три секции:

* **private** – закрытая область, доступная только внутри класса. Здесь хранятся переменные-члены, которые не должны быть изменены напрямую.
* **protected** – доступ к этой секции есть у самого класса и его потомков (наследников).
* **public** – открытая область, доступная извне. Здесь обычно находятся методы для работы с классом (геттеры, сеттеры и другие функции).

Пример простого класса в C++:

class Product **{**

private**:**

std**::**string name**;**

std**::**string manufacturer**;**

std**::**string unit**;**

double price**;**

public**:**

// Конструктор

Product**(**std**::**string name**,** std**::**string manufacturer**,** std**::**string unit**,** double price**);**

// Методы для получения информации о товаре

std**::**string getName**()** const**;**

std**::**string getManufacturer**()** const**;**

std**::**string getUnit**()** const**;**

double getPrice**()** const**;**

// Метод для отображения информации о товаре

void display**()** const**;**

**};**

2. Используя объектно-ориентированный подход (допускается STL) написать программу, выполняющую следующие действия:   
а) ввод с консоли информации о товаре (наименование, производитель, единица измерения количества, стоимость)   
б) запись в файл.   
в) поиск в файле товара с наименьшей или наибольшей (по выбору пользователя) ценой.  
Код:  
#include <iostream> // Подключаем для работы с вводом-выводом  
#include <fstream> // Подключаем для работы с файлами  
#include <vector> // Подключаем вектор (динамический массив)  
#include <limits> // Для работы с максимальными и минимальными значениямиы

// Используем пространство имен std, чтобы не писать std:: перед каждой функцией

**using** **namespace** std**;**

// Объявляем класс Product

class Product **{**

private**:**

string name**;** // Название товара

string manufacturer**;** // Производитель

string unit**;** // Единица измерения

double price**;** // Цена товара

public**:**

// Конструктор для инициализации объекта

Product**(**string name**,** string manufacturer**,** string unit**,** double price**)**

**:** name**(**name**),** manufacturer**(**manufacturer**),** unit**(**unit**),** price**(**price**)** **{}**

// Методы доступа (геттеры)

string getName**()** const **{** **return** name**;** **}**

string getManufacturer**()** const **{** **return** manufacturer**;** **}**

string getUnit**()** const **{** **return** unit**;** **}**

double getPrice**()** const **{** **return** price**;** **}**

// Метод для отображения информации о товаре

void display**()** const **{**

cout **<<** "Название: " **<<** name **<<** "\n"

**<<** "Производитель: " **<<** manufacturer **<<** "\n"

**<<** "Единица измерения: " **<<** unit **<<** "\n"

**<<** "Цена: " **<<** price **<<** " руб.\n"

**<<** "----------------------\n"**;**

**}**

// Метод для сохранения информации в файл

void saveToFile**(**ofstream**&** file**)** const **{**

file **<<** name **<<** "," **<<** manufacturer **<<** "," **<<** unit **<<** "," **<<** price **<<** "\n"**;**

**}**

// Метод для загрузки товара из строки (используется при чтении из файла)

static Product loadFromString**(**const string**&** line**)** **{**

size\_t pos1 **=** line**.**find**(**','**);**

size\_t pos2 **=** line**.**find**(**','**,** pos1 **+** 1**);**

size\_t pos3 **=** line**.**find**(**','**,** pos2 **+** 1**);**

string name **=** line**.**substr**(**0**,** pos1**);**

string manufacturer **=** line**.**substr**(**pos1 **+** 1**,** pos2 **-** pos1 **-** 1**);**

string unit **=** line**.**substr**(**pos2 **+** 1**,** pos3 **-** pos2 **-** 1**);**

double price **=** stod**(**line**.**substr**(**pos3 **+** 1**));**

**return** Product**(**name**,** manufacturer**,** unit**,** price**);**

**}**

**};**

// Функция для ввода данных о товаре

Product inputProduct**()** **{**

string name**,** manufacturer**,** unit**;**

double price**;**

cout **<<** "Введите название товара: "**;**

getline**(**cin**,** name**);**

cout **<<** "Введите производителя: "**;**

getline**(**cin**,** manufacturer**);**

cout **<<** "Введите единицу измерения (например, кг, шт, л): "**;**

getline**(**cin**,** unit**);**

cout **<<** "Введите цену: "**;**

cin **>>** price**;**

cin**.**ignore**();** // Убираем символ новой строки после cin

**return** Product**(**name**,** manufacturer**,** unit**,** price**);**

**}**

// Функция для записи списка товаров в файл

void saveProductsToFile**(**const vector**<**Product**>&** products**,** const string**&** filename**)** **{**

ofstream file**(**filename**);**

**if** **(!**file**)** **{**

cout **<<** "Ошибка открытия файла!\n"**;**

**return;**

**}**

**for** **(**const auto**&** product **:** products**)** **{**

product**.**saveToFile**(**file**);**

**}**

file**.**close**();**

cout **<<** "Товары сохранены в файл " **<<** filename **<<** "\n"**;**

**}**

// Функция для загрузки товаров из файла

vector**<**Product**>** loadProductsFromFile**(**const string**&** filename**)** **{**

vector**<**Product**>** products**;**

ifstream file**(**filename**);**

**if** **(!**file**)** **{**

cout **<<** "Ошибка открытия файла!\n"**;**

**return** products**;**

**}**

string line**;**

**while** **(**getline**(**file**,** line**))** **{**

products**.**push\_back**(**Product**::**loadFromString**(**line**));**

**}**

file**.**close**();**

**return** products**;**

**}**

// Функция для поиска товара с минимальной или максимальной ценой

void findProductByPrice**(**const vector**<**Product**>&** products**,** bool findMax**)** **{**

**if** **(**products**.**empty**())** **{**

cout **<<** "Список товаров пуст!\n"**;**

**return;**

**}**

Product bestProduct **=** products**[**0**];** // Берем первый товар как начальный

**for** **(**const auto**&** product **:** products**)** **{**

**if** **((**findMax **&&** product**.**getPrice**()** **>** bestProduct**.**getPrice**())** **||**

**(!**findMax **&&** product**.**getPrice**()** **<** bestProduct**.**getPrice**()))** **{**

bestProduct **=** product**;**

**}**

**}**

cout **<<** **(**findMax **?** "Самый дорогой товар:\n" **:** "Самый дешевый товар:\n"**);**

bestProduct**.**display**();**

**}**

// Главная функция программы

int main**()** **{**

vector**<**Product**>** products**;** // Хранение списка товаров

string filename **=** "products.txt"**;**

**while** **(true)** **{**

cout **<<** "Выберите действие:\n"

**<<** "1. Ввести новый товар\n"

**<<** "2. Сохранить товары в файл\n"

**<<** "3. Загрузить товары из файла\n"

**<<** "4. Найти товар с минимальной ценой\n"

**<<** "5. Найти товар с максимальной ценой\n"

**<<** "0. Выход\n"

**<<** "Ваш выбор: "**;**

int choice**;**

cin **>>** choice**;**

cin**.**ignore**();** // Убираем символ новой строки

**switch** **(**choice**)** **{**

**case** 1**:**

products**.**push\_back**(**inputProduct**());**

**break;**

**case** 2**:**

saveProductsToFile**(**products**,** filename**);**

**break;**

**case** 3**:**

products **=** loadProductsFromFile**(**filename**);**

**break;**

**case** 4**:**

findProductByPrice**(**products**,** **false);**

**break;**

**case** 5**:**

findProductByPrice**(**products**,** **true);**

**break;**

**case** 0**:**

**return** 0**;**

**default:**

cout **<<** "Неверный ввод, попробуйте снова!\n"**;**

**}**

**}**

**}**