# **Прикладное программирование**

Практика 4

Объектная модель Графическое перо. Сортировка в массиве объектов

Каждую программу сохранять в отдельном файле для каждого упражнения,

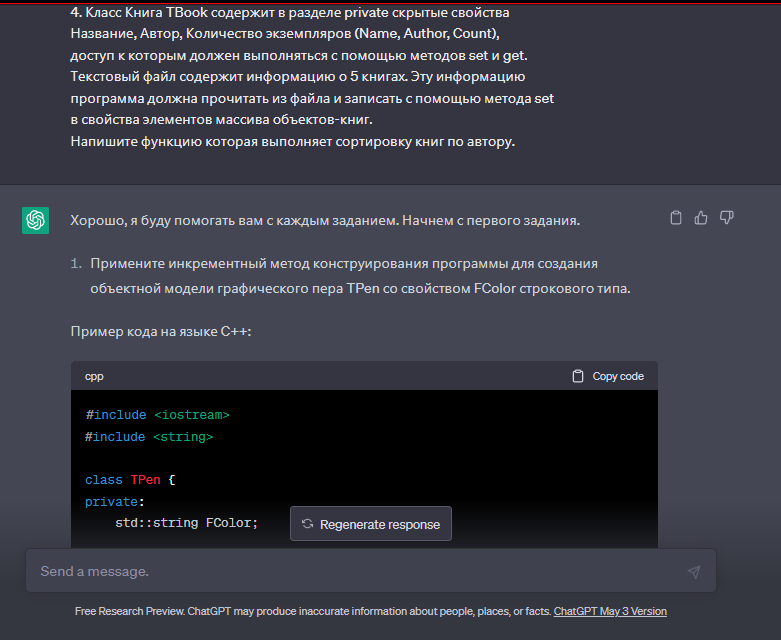
например prog04-1, prog04-2 и т.д., где "04" - номер практического

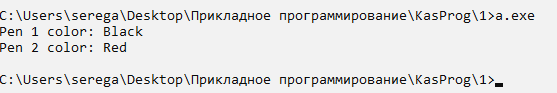
занятия, 1, 2 и т.д. - номер задания.

1. Примените инкрементный метод конструирования программы для решения следующей

задачи. По примеру 2 из лекции создайте объектную модель графического

пера TPen со свойствством FColor строкового типа.



  
#include <iostream>

#include <string>

class TPen {

private:

std::string FColor;

public:

TPen() {

FColor = "Black";

}

TPen(const std::string& color) {

FColor = color;

}

void SetColor(const std::string& color) {

FColor = color;

}

std::string GetColor() const {

return FColor;

}

};

int main() {

TPen pen1; // Создание объекта с использованием конструктора без параметров

std::cout << "Pen 1 color: " << pen1.GetColor() << std::endl;

TPen pen2("Red"); // Создание объекта с использованием конструктора с параметрами

std::cout << "Pen 2 color: " << pen2.GetColor() << std::endl;

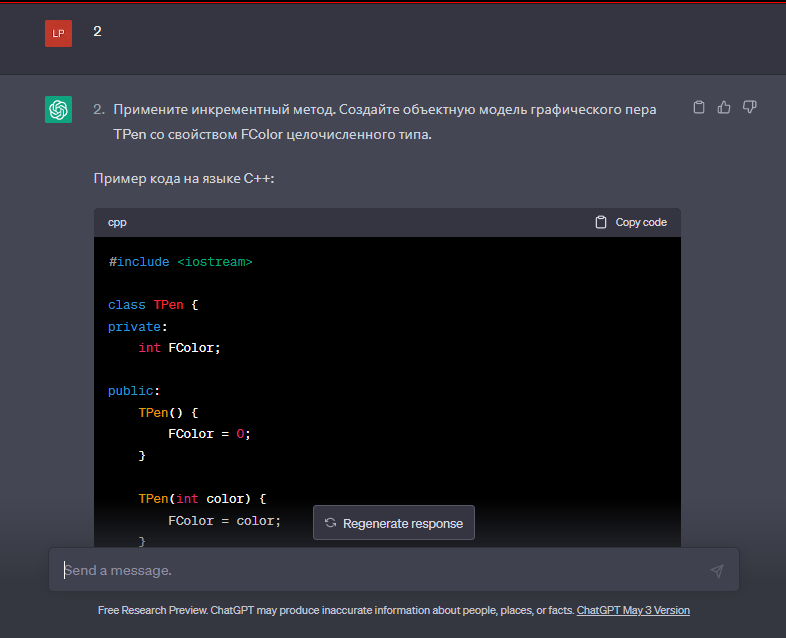
return 0;

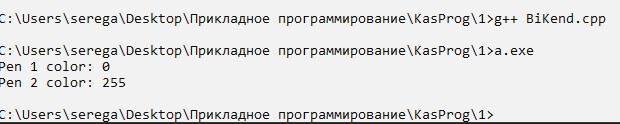
}

2. Примените инкрементный метод. По примеру 2 из лекции создайте

объектную модель графического пера TPen со свойствством FColor

целочисленного типа.





#include <iostream>

class TPen {

private:

int FColor;

public:

TPen() {

FColor = 0;

}

TPen(int color) {

FColor = color;

}

void SetColor(int color) {

FColor = color;

}

int GetColor() const {

return FColor;

}

};

int main() {

TPen pen1; // Создание объекта с использованием конструктора без параметров

std::cout << "Pen 1 color: " << pen1.GetColor() << std::endl;

TPen pen2(255); // Создание объекта с использованием конструктора с параметрами

std::cout << "Pen 2 color: " << pen2.GetColor() << std::endl;

return 0;

}

3. Составьте программный модуль с помощью инкрементного метода.

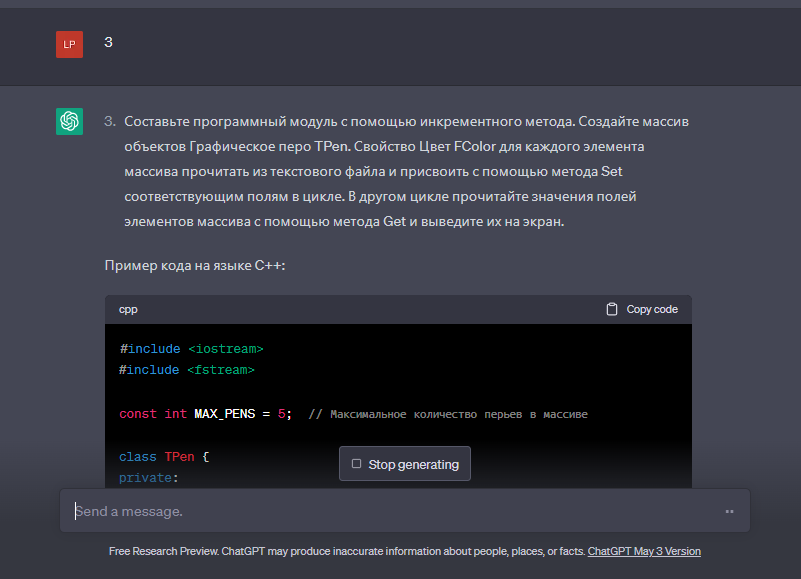
Создайте массив объектов Графическое перо TPen. Свойство Цвет

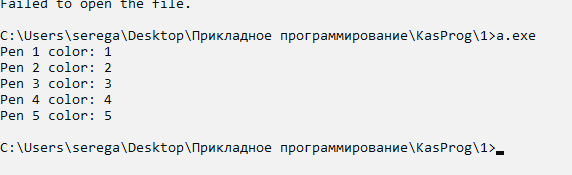
FColor для каждого элемента массива прочитать из текстового

файла и присвоить с помощью метода set соответствующим полям

в цикле. В другом цикле прочитать значения полей элементов

массива с помощью метода get и вывести на экран.





#include <iostream>

#include <fstream>

const int MAX\_PENS = 5; // Максимальное количество перьев в массиве

class TPen {

private:

int FColor;

public:

TPen() {

FColor = 0;

}

void SetColor(int color) {

FColor = color;

}

int GetColor() const {

return FColor;

}

};

int main() {

TPen pens[MAX\_PENS]; // Массив объектов класса TPen

std::ifstream inputFile("colors.txt"); // Открытие текстового файла для чтения

if (!inputFile) {

std::cout << "Failed to open the file." << std::endl;

return 1;

}

// Заполнение массива перьев значениями цветов из файла

for (int i = 0; i < MAX\_PENS; i++) {

int color;

inputFile >> color; // Чтение значения цвета из файла

pens[i].SetColor(color); // Присвоение цвета элементу массива

}

inputFile.close(); // Закрытие файла

// Вывод значений цветов перьев на экран

for (int i = 0; i < MAX\_PENS; i++) {

std::cout << "Pen " << i+1 << " color: " << pens[i].GetColor() << std::endl;

}

return 0;

}

4. Класс Книга TBook содержит в разделе private скрытые свойства

Название, Автор, Количество экземпляров (Name, Author, Count),

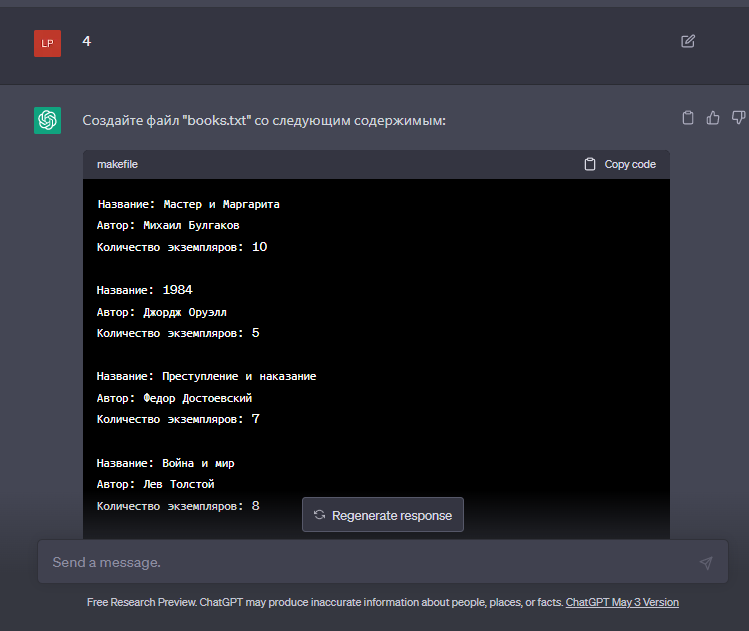
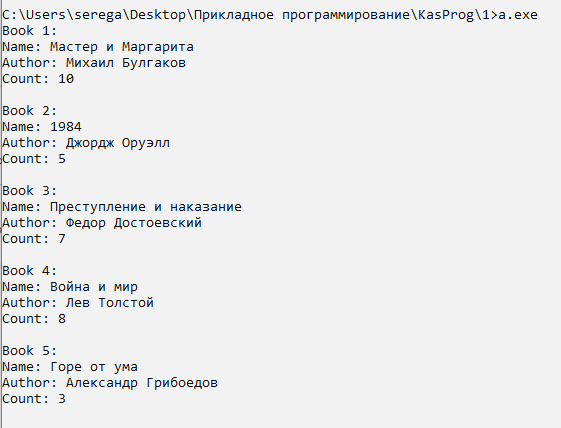
доступ к которым должен выполняться с помощью методов set и get.

Текстовый файл содержит информацию о 5 книгах. Эту информацию

программа должна прочитать из файла и записать с помощью метода set

в свойства элементов массива объектов-книг.

Напишите функцию которая выполняет сортировку книг по автору.

  
  
#include <iostream>

#include <fstream>

#include <string>

using namespace std;

class TBook {

private:

string Name;

string Author;

int Count;

public:

void SetName(const string& name) {

Name = name;

}

string GetName() const {

return Name;

}

void SetAuthor(const string& author) {

Author = author;

}

string GetAuthor() const {

return Author;

}

void SetCount(int count) {

Count = count;

}

int GetCount() const {

return Count;

}

};

int main() {

const int NUM\_BOOKS = 5;

TBook books[NUM\_BOOKS];

ifstream inputFile("books.txt");

if (!inputFile) {

cout << "Failed to open file." << endl;

return 1;

}

for (int i = 0; i < NUM\_BOOKS; i++) {

string line;

getline(inputFile, line);

string name = line.substr(line.find(':') + 2);

books[i].SetName(name);

getline(inputFile, line);

string author = line.substr(line.find(':') + 2);

books[i].SetAuthor(author);

getline(inputFile, line);

int count = stoi(line.substr(line.find(':') + 2));

books[i].SetCount(count);

getline(inputFile, line); // Read the empty line between books

}

inputFile.close();

// Print the book information

for (int i = 0; i < NUM\_BOOKS; i++) {

cout << "Book " << i + 1 << ":" << endl;

cout << "Name: " << books[i].GetName() << endl;

cout << "Author: " << books[i].GetAuthor() << endl;

cout << "Count: " << books[i].GetCount() << endl;

cout << endl;

}

return 0;

}