# **Прикладное программирование**

Практика 5

Библиотеки: класс логических элементов, класс Книга

Для каждой программы, состоящей из нескольких файлов,

создавайте отдельную папку проекта, например project05-1,

project05-2 и т.д., где "05" - номер практического

занятия, 1, 2 и т.д. - номер задания.

Каждую программу, состоящую из одного файла, сохранять

в отдельном файле для каждого упражнения,

например prog05-1, prog05-2 и т.д., где "05" - номер практического

занятия, 1, 2 и т.д. - номер задания.

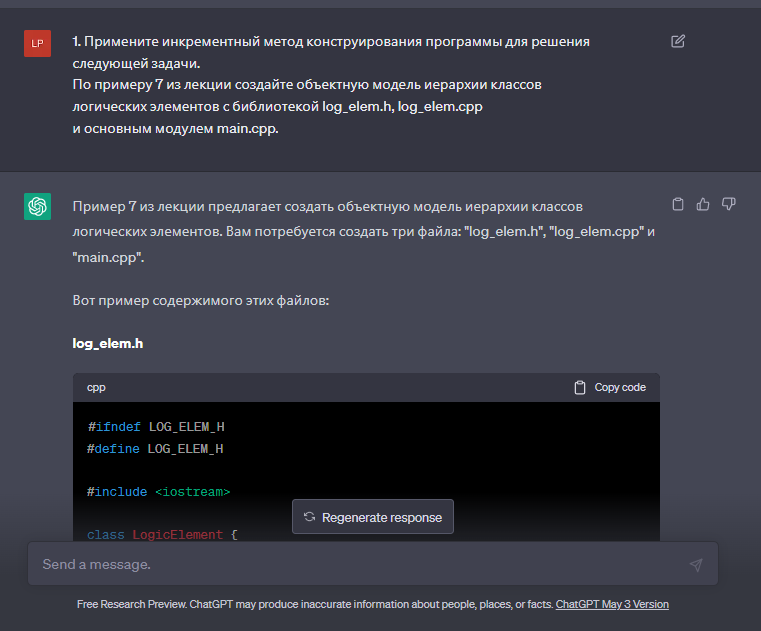
1. Примените инкрементный метод конструирования программы для решения следующей задачи.

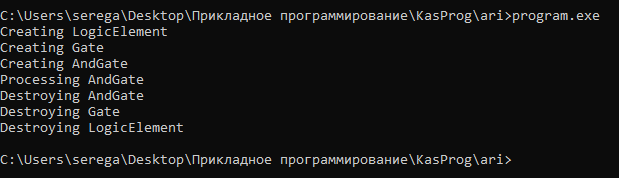
По примеру 7 из лекции создайте объектную модель иерархии классов

логических элементов с библиотекой log\_elem.h, log\_elem.cpp

и основным модулем main.cpp.

Оценка: 1) 1 балл за каждую версию-инкремент.



  
log\_elem.h  
#ifndef LOG\_ELEM\_H

#define LOG\_ELEM\_H

#include <iostream>

class LogicElement {

public:

LogicElement() {

std::cout << "Creating LogicElement" << std::endl;

}

virtual ~LogicElement() {

std::cout << "Destroying LogicElement" << std::endl;

}

virtual void process() = 0;

};

class Gate : public LogicElement {

public:

Gate() {

std::cout << "Creating Gate" << std::endl;

}

~Gate() {

std::cout << "Destroying Gate" << std::endl;

}

void process() override {

std::cout << "Processing Gate" << std::endl;

}

};

class AndGate : public Gate {

public:

AndGate() {

std::cout << "Creating AndGate" << std::endl;

}

~AndGate() {

std::cout << "Destroying AndGate" << std::endl;

}

void process() override {

std::cout << "Processing AndGate" << std::endl;

}

};

#endif

log\_elem.cpp  
#include "log\_elem.h"

// Здесь можно реализовать необходимые методы, если требуется.

Main.cpp  
#include "log\_elem.h"

int main() {

LogicElement\* element = new AndGate();

element->process();

delete element;

return 0;

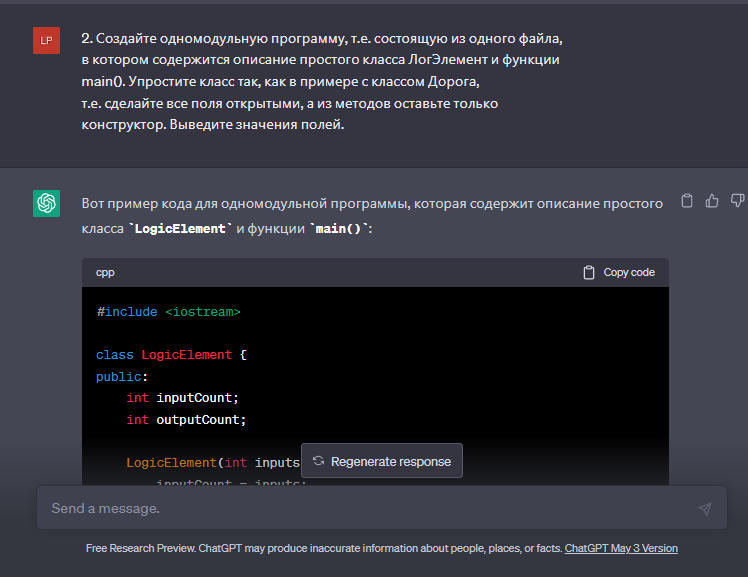
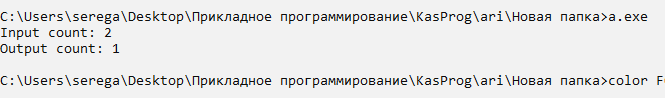
}

2. Создайте одномодульную программу, т.е. состоящую из одного файла,

в котором содержится описание простого класса ЛогЭлемент и функции

main(). Упростите класс так, как в примере с классом Дорога,

т.е. сделайте все поля открытыми, а из методов оставьте только

конструктор. Выведите значения полей.  
  
  
#include <iostream>

class LogicElement {

public:

int inputCount;

int outputCount;

LogicElement(int inputs, int outputs) {

inputCount = inputs;

outputCount = outputs;

}

};

int main() {

LogicElement element(2, 1);

std::cout << "Input count: " << element.inputCount << std::endl;

std::cout << "Output count: " << element.outputCount << std::endl;

return 0;

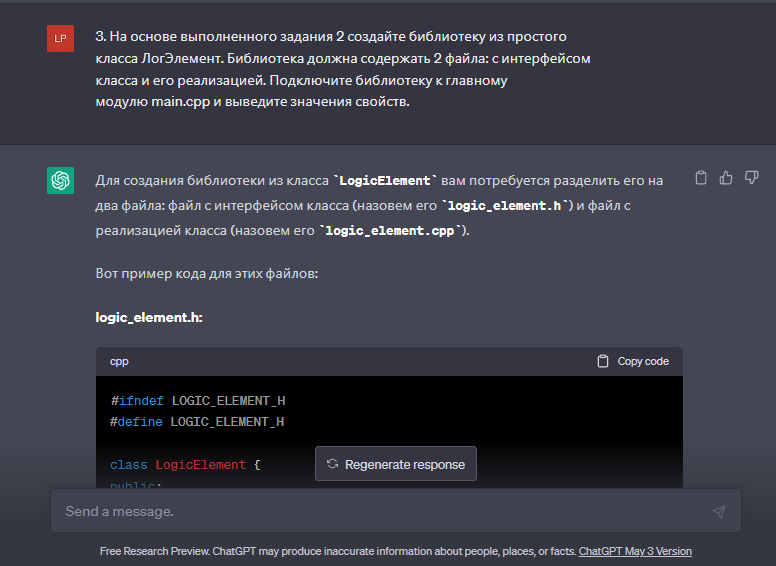
}

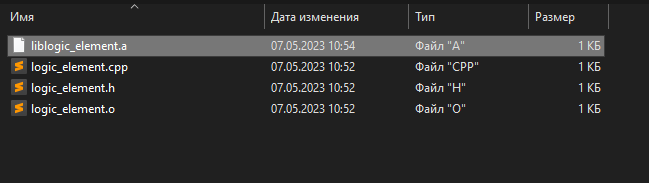
3. На основе выполненного задания 2 создайте библиотеку из простого

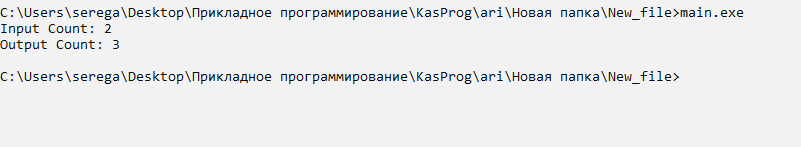
класса ЛогЭлемент. Библиотека должна содержать 2 файла: с интерфейсом

класса и его реализацией. Подключите библиотеку к главному

модулю main.cpp и выведите значения свойств.





  
#include "logic\_element.h"

LogicElement::LogicElement(int inputs, int outputs) {

inputCount = inputs;

outputCount = outputs;

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
#ifndef LOGIC\_ELEMENT\_H

#define LOGIC\_ELEMENT\_H

class LogicElement {

public:

int inputCount;

int outputCount;

LogicElement(int inputs, int outputs);

};

#endif

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4c01 0500 0000 0000 6201 0000 0f00 0000

0000 0401 2e74 6578 7400 0000 0000 0000

0000 0000 2000 0000 dc00 0000 0000 0000

0000 0000 0000 0000 2000 3060 2e64 6174

6100 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

4000 30c0 2e62 7373 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

0000 0000 0000 0000 8000 30c0 2f34 0000

0000 0000 0000 0000 0000 0000 2400 0000

fc00 0000 0000 0000 0000 0000 0000 0000

4000 3040 2f31 3500 0000 0000 0000 0000

0000 0000 3800 0000 2001 0000 5801 0000

0000 0000 0100 0000 4000 3040 5589 e583

ec04 894d fc8b 45fc 8b55 0889 108b 45fc

8b55 0c89 5004 90c9 c208 0090 4743 433a

2028 4d69 6e47 572e 6f72 6720 4743 4320

4275 696c 642d 3229 2039 2e32 2e30 0000

1400 0000 0000 0000 017a 5200 017c 0801

1b0c 0404 8801 0000 1c00 0000 1c00 0000

0400 0000 1f00 0000 0041 0e08 8502 420d

0559 c50c 0404 0000 2000 0000 0500 0000

1400 2e66 696c 6500 0000 0000 0000 feff

0000 6701 6c6f 6769 635f 656c 656d 656e

742e 6370 7000 0000 0000 1900 0000 0000

0000 0100 2000 0201 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 0000 3100

0000 0000 0000 0100 2000 0200 2e74 6578

7400 0000 0000 0000 0100 0000 0301 1f00

0000 0000 0000 0000 0000 0000 0000 0000

2e64 6174 6100 0000 0000 0000 0200 0000

0301 0000 0000 0000 0000 0000 0000 0000

0000 0000 2e62 7373 0000 0000 0000 0000

0300 0000 0301 0000 0000 0000 0000 0000

0000 0000 0000 0000 0000 0000 4900 0000

0000 0000 0400 0000 0301 2300 0000 0000

0000 0000 0000 0000 0000 0000 0000 0000

5400 0000 0000 0000 0500 0000 0301 3800

0000 0100 0000 0000 0000 0000 0000 0000

5e00 0000 2e72 6461 7461 247a 7a7a 002e

6568 5f66 7261 6d65 005f 5f5a 4e31 324c

6f67 6963 456c 656d 656e 7443 3245 6969

005f 5f5a 4e31 324c 6f67 6963 456c 656d

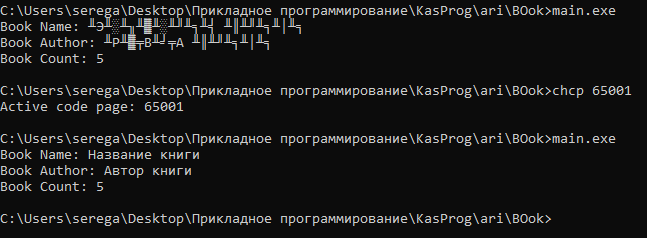
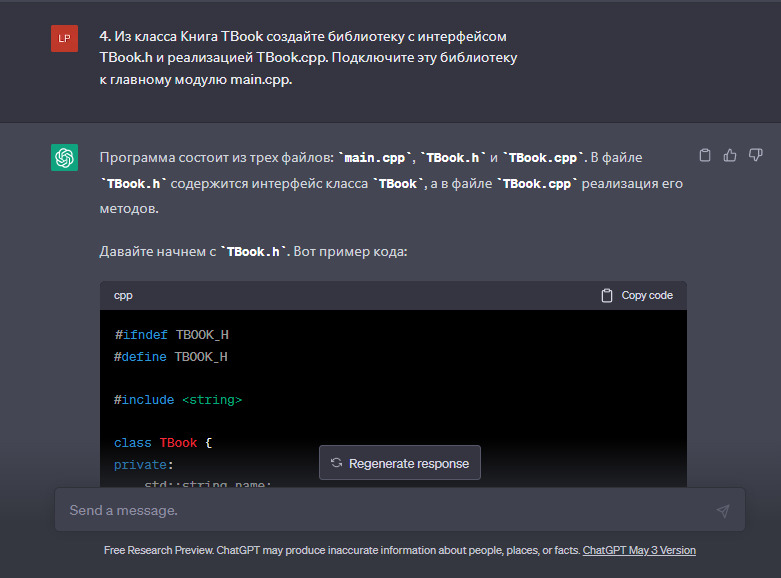
656e 7443 3145 6969 002e 7264 6174 6124

7a7a 7a00 2e65 685f 6672 616d 6500

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
4. Из класса Книга TBook создайте библиотеку с интерфейсом

TBook.h и реализацией TBook.cpp. Подключите эту библиотеку

к главному модулю main.cpp.

  
#ifndef TBOOK\_H

#define TBOOK\_H

#include <string>

class TBook {

private:

std::string name;

std::string author;

int count;

public:

TBook(const std::string& name, const std::string& author, int count);

std::string getName() const;

std::string getAuthor() const;

int getCount() const;

void setName(const std::string& name);

void setAuthor(const std::string& author);

void setCount(int count);

};

#endif

—--------------------------------------------------------------------------------------------------------------  
#include "TBook.h"

TBook::TBook(const std::string& name, const std::string& author, int count)

: name(name), author(author), count(count) {}

std::string TBook::getName() const {

return name;

}

std::string TBook::getAuthor() const {

return author;

}

int TBook::getCount() const {

return count;

}

void TBook::setName(const std::string& name) {

this->name = name;

}

void TBook::setAuthor(const std::string& author) {

this->author = author;

}

void TBook::setCount(int count) {

this->count = count;

}

—--------------------------------------------------------------------------------------------------------------  
#include <iostream>

#include "TBook.h"

int main() {

TBook book("Название книги", "Автор книги", 5);

std::cout << "Book Name: " << book.getName() << std::endl;

std::cout << "Book Author: " << book.getAuthor() << std::endl;

std::cout << "Book Count: " << book.getCount() << std::endl;

return 0;

}