|  |  |
| --- | --- |
| для прик эмбл | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Московский технологический университет"МИРЭА | |
| Факультет информационных технологий (ИТ) | |
| Кафедра практической и прикладной информатики | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | |
| **по дисциплине** | |
| **«**Объектно-ориентированное программирование**»** | |
|  | |
| Выполнил студент группы ИНБО-04-18 | Дмитриев А.Р. |
| Принял ассистент кафедры | Хлебникова В.Л. |

|  |  |  |
| --- | --- | --- |
| Практические работы выполнены | «25» марта 2019 г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |

Москва 2019

**Таблица успеваемости**

**Практические занятия**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| №/р. | страница в отчете. | Студент выполнил:  (подпись) | Преподаватель принял:  (подпись) | Балл  (максимально возможный) | Балл (Фактический) |
| 1. |  |  |  | 2 | … |
| 2. |  |  |  | 2 |  |
| 3. |  |  |  | 4 |  |
| 4. |  |  |  | 6 |  |
| 5. |  |  |  | 6 |  |
| 6. |  |  |  | 8 |  |
| 7. |  |  |  | 8 |  |
|  |  |  | Итоговый балл: | 36 |  |

**Практическая работа №1**

**Изучение синтаксиса языка C++**

**Цель практической работы**

Целью данной практической работы является знакомство с синтаксисом, основными алгоритмическими конструкциями языка С++, а также приобретение практических навыков разработки объектно-ориентированных программ на языке программирования С++.

**Задачи**

1. Реализовать программу, которая считывает имя и здоровается
2. Реализовать программу, производящую операции над числами. (Сложение/вычитание, целые числа/числа с плавающей запятой)

**Ход работы**

Программа 1

#include <iostream>

#include <string>

using namespace std;

int main()

{

string name;// Строковая переменная

cout << "Tell me your name.\n";

cin >> name; //Ввод имени

cout << "Hello, " << name << "!" << endl;//Приветствие

system("pause");

}

Программа 2

#include <iostream>

#include <string>

using namespace std;

int main()

{

float a, b; //Числа с плавающей точкой

string znak; //Строковая переменная

cout << "Enter first number:"; //Ввод примера

cin >> a;

cout << "Enter operation(+ , -):";

cin >> znak;

cout << "Enter second number:";

cin >> b;

cout << "Result:"; //Результат вычислений

if (znak == "+")

cout << a + b;

if (znak == "-")

cout << a - b;

system("pause");

}

**Вывод**

В ходе выполнения практической работы была использована библиотека стандартного ввода и вывода, операторы ввода, вывода, сложения и вычитания. Исходный код программ также доступен на GitHub по ссылке:

<https://github.com/WarerBit/OOP/tree/master/P1>

**Практическая работа №1**

**Решение задач на нелинейные алгоритмические конструкции языка С++**

**Цель практической работы**

Целью данной практической работы является получение практических навыков по разработке нелинейных алгоритмов с использованием конструкции «if-else».

**Ход работы**

Программа 1

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float a, b, n, m;

cout << "Введите 2 числа:"; //Ввод чисел

cin >> a >> b;

if (a >= b) //Проверка,вычисление n в соответсвии с результатом проверки

n = pow((a - b), 1 / 3.0);

else

n = a \* a + (a - b) / (sin(a\*b));

if (n < b) //Проверка,вычисление m в соответсвии с результатом проверки

m = (n + a) / (-b) + sqrt(sin(a)\*sin(a) - cos(n));

else

if (n == b)

m = b \* b + tan(n\*a);

else

if (n > b)

m = b \* b\*b + n \* a\*a;

cout << "a = " << a << "\nb = " << b << "\nn = " << n << "\nm = " << m;//Вывод результата

system("pause");

return 0;

}

Программа 2

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float a, b, z, t;

cout << "Введите 2 числа:"; //Ввод чисел

cin >> a >> b;

if (a < b) //Проверка,вычисление z в соответсвии с результатом проверки

z = sqrt(abs(a\*a - b \* b));

else

z = 1 - 2 \* cos(a)\*sin(b);

if (z < b)//Проверка,вычисление t в соответсвии с результатом проверки

t = pow((z + a \* a\*b), 1 / 3.0);

else

if (z == b)

t = 1 - log10(z) + cos(a\*a\*b);

else

if (z > b)

t = 1 / cos(z\*a);

cout << "a = " << a << "\nb = " << b << "\nz = " << z << "\nt = " << t; // Вывод результата

return 0;

}

Программа 3

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float a, b, y, t;

cout << "Введите 2 числа:";//Ввод чисел

cin >> a >> b;

if (a <= b)//Проверка,вычисление y в соответсвии с результатом проверки

y = ((a - b)\*(a + b)) / ((a + b)\*(a\*a - a \* b + b \* b));

else

y = a + log(b\*b);

if (y == b)//Проверка,вычисление t в соответсвии с результатом проверки

t = (2 \* y + sqrt(y\*y - a)) / (2 \* b - sqrt(a\*a - y));

else

if (y < b)

t = sin(y)\*sin(y) + 1 / tan(a - b);

else

if (y > b)

t = cbrt(y\*sin(a)) + 1 / sqrt(y\*cos(b));

cout << "a = " << a << "\nb = " << b << "\ny = " << y << "\nt = " << t;//Вывод результата

return 0;

}

Программа 4

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

float a, x, y, t;

cout << "Введите 2 числа:";//Ввод чисел

cin >> a >> x;

if (a <= x)//Проверка,вычисление y в соответсвии с результатом проверки

y = a + log(x + a);

else

y = sqrt(sin(a\*x));

if (a > y)//Проверка,вычисление t в соответсвии с результатом проверки

t = y / (a - x);

else

if (a == y)

t = y / (a - x) + (a + x) / (y\*y);

else

if (a < y)

t = tan(a\*x) + cos(2 \* a\*y);

cout << "a = " << a << "\nx = " << x << "\ny = " << y << "\nt = " << t;//Вывод результата

return 0;

}

**Вывод**

В ходе выполнения практической работы была использована библиотека cmath, которая позволила вычислять сложные математические функции, а также быстро совершить проверку на ошибки, и конструкция ветвления «if-else». Исходный код программ также доступен на GitHub по ссылке:

<https://github.com/WarerBit/OOP/tree/master/P2>

**Практическая работа №3**

**Классы в языке С++**

**Цель практической работы**

Целью данной практической работы является приобретение практических навыков при разработке классов на языке программирования С++.

**Задачи**

1. Реализовать класс «Книга»
2. Реализовать класс «Группа студентов»
3. Реализовать класс «Автомобиль»
4. Реализовать класс «Вектор»
5. Реализовать класс «Библиотека»

**Ход работы**

Программа 1  
Файл Book.h  
#pragma once

#include <string>

#define B\_AUTHOR 0

#define B\_TITLE 1

#define B\_PAGE 2

using namespace std;

class Book

{

string author;

string title;

int page;

public:

Book();//Конструктор

void setBookValue(int name, string value);//Задать значение author,title.

void setBookValue(int name, int value);//Задать значение page.

string getBookValue(int name);//Получить значение author,title.

int getBookPage();//Получить значение page.

~Book();//Деструктор

};

Файл Book.cpp  
#include "Book.h"

Book::Book()

{

author = "";

title = "";

page = 0;

}

void Book::setBookValue(int name, string value) {

if (name == B\_AUTHOR)

author = value;

if (name == B\_TITLE)

title = value;

}

void Book::setBookValue(int name, int value) {

page = value;

}

string Book::getBookValue(int name) {

if (name == B\_AUTHOR)

return author;

if (name == B\_TITLE)

return title;

}

int Book::getBookPage() {

return page;

}

Book::~Book()

{

}

Файл Source.cpp  
#include "Book.h"

#include <iostream>

#include <ctime>

int main()

{

Book kniga;

string vvod;

int vod;

cout << "Vvedite avtora:";

cin >> vvod;

kniga.setBookValue(B\_AUTHOR, vvod);

cout << "Vvedite nazvanie:";

cin >> vvod;

kniga.setBookValue(B\_TITLE, vvod);

cout << "Vvedite stranicu:";

cin >> vod;

kniga.setBookValue(B\_PAGE, vod);

cout << kniga.getBookValue(B\_AUTHOR) << " " << kniga.getBookValue(B\_TITLE) << " " << kniga.getBookPage() << endl;

system("pause");

}

Программа 2

Файл StudentGroup.h  
#pragma once

#include <string>

#define SG\_CODE 0

#define SG\_COURSE 1

using namespace std;

class StudentGroup

{

string code;

int course;

public:

StudentGroup();//Конструктор

void setSGValue(string value);//Задать значение code.

void setSGValue(int value);//Задать значение course.

string getSGCode();//Получить значение code.

int getSGCourse();//Получить значение course.

~StudentGroup();//Деструктор

};

Файл StudentGroup.cpp  
#include "StudentGroup.h"

StudentGroup::StudentGroup()

{

}

void StudentGroup::setSGValue(string value) {

code = value;

}

void StudentGroup::setSGValue(int value) {

course = value;

}

string StudentGroup::getSGCode() {

return code;

}

int StudentGroup::getSGCourse() {

return course;

}

StudentGroup::~StudentGroup()

{

}

Файл Source.cpp  
#include "StudentGroup.h"

#include <iostream>

int main()

{

StudentGroup group;

string vvod; int vod;

cout << "Vvedite cod gruppi:";

cin >> vvod;

group.setSGValue(vvod);

cout << "Vvedite curs gruppi:";

cin >> vod;

group.setSGValue(vod);

cout << "Code:" << group.getSGCode() << endl << "Course:" << group.getSGCourse() << endl;

system("pause");

}  
  
  
Программа 3

Файл Car.h  
#pragma once

#include <string>

#define C\_MODEL 0

#define C\_SEAT 1

#define C\_COLOR 2

using namespace std;

class Car

{

string model;

int seat;

string color;

public:

Car();//Конструктор

void setCarValue(int name, string value);//Задать значение model,color.

void setCarValue(int name, int value);//Задать значение seat.

string getCarValue(int name);//Получить значение model,color.

int getCarSeat();//Получить значение seat.

~Car();//Деструктор

};

Файл Car.cpp  
#include "Car.h"

Car::Car()

{

model = "";

color = "";

seat = 0;

}

void Car::setCarValue(int name, string value) {

if (name == C\_MODEL)

model = value;

if (name == C\_COLOR)

color = value;

}

void Car::setCarValue(int name, int value) {

if (name == C\_SEAT)

seat = value;

}

string Car::getCarValue(int name) {

if (name == C\_MODEL)

return model;

if (name == C\_COLOR)

return color;

}

int Car::getCarSeat() {

return seat;

}

Car::~Car()

{

}

Файл Source.cpp  
#include "Car.h"

#include <iostream>

int main()

{

Car mashina;

string vvod;

int vod;

cout << "Vvedite model:";

cin >> vvod;

mashina.setCarValue(C\_MODEL, vvod);

cout << "Vvedite kol-vo mest:";

cin >> vod;

mashina.setCarValue(C\_SEAT, vod);

cout << "Vvedite cvet:";

cin >> vvod;

mashina.setCarValue(C\_COLOR, vvod);

cout << "Model:" << mashina.getCarValue(C\_MODEL) << "\nKol-vo mest:" << mashina.getCarSeat() << "\nCvet:" << mashina.getCarValue(C\_COLOR);

system("pause");

}

Программа 4

Файл Vector.h  
#pragma once

#define V\_X 1

#define V\_Y 2

#define V\_Z 3

class Vector

{

double x;

double y;

double z;

public:

Vector();//Конструктор

void setVectorCoord(double x, double y, double z);//Задать координаты вектора.

double getVectorCoord(int name);//Получить координату вектора.

void sumVector(Vector sum);//Сложить векторы

~Vector();//Деструктор

};  
  
Файл Vector.cpp  
#include "Vector.h"

Vector::Vector()

{

x = 0;

y = 0;

z = 0;

}

void Vector::setVectorCoord(double x, double y, double z) {

this->x = x;

this->y = y;

this->z = z;

}

double Vector::getVectorCoord(int name) {

if (name == V\_X)

return x;

if (name == V\_Y)

return y;

if (name == V\_Z)

return z;

}

void Vector::sumVector(Vector sum) {

x += sum.getVectorCoord(V\_X);

y += sum.getVectorCoord(V\_Y);

z += sum.getVectorCoord(V\_Z);

}

Vector::~Vector()

{}

Файл Source.cpp  
#include "Vector.h"

#include <iostream>

using namespace std;

int main()

{

Vector strelka, napravotrezok;

double x, y, z, coord;

cout << "Vvedite coordinati vectora:";

cin >> x >> y >> z;

strelka.setVectorCoord(x, y, z);

cout << "Vvedite coordinati dr vectora:";

cin >> x >> y >> z;

napravotrezok.setVectorCoord(x, y, z);

cout << "Coordinati 1go vectora: " << strelka.getVectorCoord(V\_X) << " " << strelka.getVectorCoord(V\_Y) << " " << strelka.getVectorCoord(V\_Z) << endl;

strelka.sumVector(napravotrezok);

cout << "Coordinati vectora 1+2: " << strelka.getVectorCoord(V\_X) << " " << strelka.getVectorCoord(V\_Y) << " " << strelka.getVectorCoord(V\_Z) << endl;

system("pause");

}

Программа 5  
Файл Book.h  
#pragma once

#include <string>

#define B\_AUTHOR 0

#define B\_TITLE 1

#define B\_PAGE 2

using namespace std;

class Book

{

string author;

string title;

int page;

public:

Book();//Конструктор

void setBookValue(int name, string value);//Задать значение author,title.

void setBookValue(int name, int value);//Задать значение page.

string getBookValue(int name);//Получить значение author,title.

int getBookPage();//Получить значение page.

~Book();//Деструктор

};

Файл Book.cpp  
#include "Book.h"

Book::Book()

{

author = "";

title = "";

page = 0;

}

void Book::setBookValue(int name, string value) {

if (name == B\_AUTHOR)

author = value;

if (name == B\_TITLE)

title = value;

}

void Book::setBookValue(int name, int value) {

page = value;

}

string Book::getBookValue(int name) {

if (name == B\_AUTHOR)

return author;

if (name == B\_TITLE)

return title;

}

int Book::getBookPage() {

return page;

}

Book::~Book()

{

}

Файл Lib.h  
#pragma once

#include <vector>

#include <iostream>

#include "Book.h"

using namespace std;

class Lib

{

int shelfHeight;

int shelfWidth;

vector<Book> books;

public:

Lib();//Конструктор

void libCreate(int H, int W);//Задать значения shelfHeight и shelfWidth.

void addBook(Book book);//Добавить книгу в библиотеку

void getBook(int H, int W);//Получить книгу

void delBook(int H, int W);//Удалить книгу

~Lib();//Деструктор

};  
  
Файл Lib.cpp  
#include "Lib.h"

#include <Windows.h>

Lib::Lib()

{

shelfHeight = 0;

shelfWidth = 0;

books.clear();

}

void Lib::libCreate(int H, int W) {

shelfHeight = H;

shelfWidth = W;

books.reserve(H\*W);

}

void Lib::addBook(Book book) {

if (books.size() != shelfHeight \* shelfWidth)

if ((book.getBookValue(B\_AUTHOR) != "") && (book.getBookValue(B\_TITLE) != "") && (book.getBookPage() != 0))

books.push\_back(book);

else

cout << "Can't add this book.\n";

else

cout << "No more room in the shelf!\n";

}

void Lib::getBook(int H, int W) {

if ((H != 0) && (H <= shelfHeight) && (W != 0) && (W <= shelfWidth))

{

int place = (H - 1)\*(shelfWidth)+W - 1;

if (books.size() > place) {

cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << "Author:" << books[place].getBookValue(B\_AUTHOR) << endl;

cout << "Title:" << books[place].getBookValue(B\_TITLE) << endl;

cout << "Pages:" << books[place].getBookPage() << endl;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

}

else

cout << "No book.\n";

}

else

cout << "Wrong height or width!\n";

}

void Lib::delBook(int H, int W) {

if ((H != 0) && (H <= shelfHeight) && (W != 0) && (W <= shelfWidth))

{

int place = (H - 1)\*(shelfWidth)+W - 1;

if (books.size() > place) {

getBook(H, W);

books.erase(books.begin() + place);

cout << "\nThis book was deleted.\n";

}

else

cout << "No book.\n";

}

else

cout << "Wrong height or width!\n";

}

Lib::~Lib()

{

}  
  
Файл Source.cpp  
#include "Lib.h"

int main()

{

Lib shelf;

Book book;

int h, w;

string s;

cout << "Vvedite kolichestvo polok i shirinu polki:";

cin >> h >> w;

shelf.libCreate(h, w);

for (int i = 0; i < 3; i++)

{

cout << "Dobavte knigu:\n";

cout << "Author:";

cin >> s;

book.setBookValue(B\_AUTHOR, s);

cout << "Title:";

cin >> s;

book.setBookValue(B\_TITLE, s);

cout << "Pages:";

cin >> h;

book.setBookValue(B\_PAGE, h);

shelf.addBook(book);

}

cout << "Vvedite nomer polki i nomer knigi,kotoruu hotite uvidet:";

cin >> h >> w;

shelf.getBook(h, w);

cout << "Vvedite nomer polki i nomer knigi,kotoruu hotite udalit:";

cin >> h >> w;

shelf.delBook(h, w);

system("pause");

}

**Вывод**

Все поставленные задачи были выполнены. Для этого были использованы библиотека STL,шаблоны функций. Все задачи были выполнены в виде много файлового проекта с разбиением классов на файлы объявления и реализации класса. Исходный код программ также доступен на GitHub по ссылке:

<https://github.com/WarerBit/OOP/tree/master/P3>

**Практическая работа №4**   
**Динамическое выделение памяти для объектов в языке С++**

**Цель практической работы**

Целью данной практической работы является изучение динамического выделения и удаления памяти, а так же работа с указателями на языке С++.

**Задачи**

1. Объявите указатель на массив типа double и предложите пользователю выбрать его размер. Далее напишите четыре функции: первая должна выделить память для массива, вторая – заполнить ячейки данными, третья – показать данные на экран, четвёртая – освободить занимаемую память. Программа должна предлагать продолжать работу (создавать новые массивы) или выйти из программы.
2. Объявите указатель на массив типа int и выделите память для 12-ти элементов. Необходимо написать функцию, которая поменяет значения четных и нечетных ячеек массива.
3. Объявить и заполнить двумерный динамический массив случайными числами от 10 до 50. Показать его на экран. Для заполнения и показа на экран написать отдельные функции. Количество строк и столбцов выбирает пользователь

**Ход работы**

Программа 1

#include <iostream>

#include <ctime>

using namespace std;

double \*cMem(int n);//Создает массив

void filArr(double \*a, int n);//Заполняет массив случайными числами

void displayArr(double \*a, int n);//Выводит массив на экран

void delArr(double \*a);//Удаляет массив

int main()

{

srand(time(NULL));

int n;

cout << "Vvedite razmer massiva:";

cin >> n;

double \*a;

a = cMem(n);

filArr(a, n);

displayArr(a, n);

delArr(a);

system("pause");

}

double \*cMem(int n) {

double \*a = new double[n];

return a;

}

void filArr(double \*a, int n) {

for (int i = 0; i < n; i++)

a[i] = (rand() % 1000) / 7.0;

}

void displayArr(double \*a, int n) {

for (int i = 0; i < n; i++)

cout << "element #" << i + 1 << " = " << a[i] << endl;

}

void delArr(double \*a) {

delete[] a;

}

Программа 2  
  
#include <iostream>

#include <ctime>

using namespace std;

void filArr(int \*a);//Заполняет массив случайными числами

void changeArr(int \*a);//Меняет местами пары элементов массива

void displayArr(int \*a);//Выводит массив на экран

void delArr(int \*a);//Удаляет массив

int main()

{

srand(time(NULL));

int \*a = new int[12];

filArr(a);

cout << "Nachalniy massiv:\n";

displayArr(a);

changeArr(a);

cout << "\n\nKonechniy massiv:\n";

displayArr(a);

delArr(a);

system("pause");

}

void changeArr(int \*a) {

for (int i = 0; i < 11; i += 2)

swap(a[i], a[i + 1]);

}

void filArr(int \*a) {

for (int i = 0; i < 12; i++)

a[i] = (rand() % 100);

}

void displayArr(int \*a) {

for (int i = 0; i < 12; i++)

cout << "element #" << i + 1 << " = " << a[i] << endl;

}

void delArr(int \*a) {

delete[] a;

}  
  
  
Программа 3  
  
#include <iostream>

#include <ctime>

using namespace std;

void filArr(int \*\*a, int i, int j);//Заполняет массив случайными числами

void displayArr(int \*\*a, int i, int j);//Выводит массив на экран

void delArr(int \*\*a, int i);//Удаляет массив

int main()

{

srand(time(NULL));

int i, j;

cout << "Vvedite kol-vo strok i stolbcov:";

cin >> i >> j;

int \*\*a = new int\*[i];

for (int k = 0; k < i; k++)

a[k] = new int[j];

filArr(a, i, j);

displayArr(a, i, j);

delArr(a, i);

system("pause");

}

void filArr(int \*\*a, int i, int j) {

for (int k = 0; k < i; k++)

for (int t = 0; t < j; t++)

a[k][t] = (rand() % 40) + 10;

}

void displayArr(int \*\*a, int i, int j) {

for (int k = 0; k < i; k++) {

cout << endl;

for (int t = 0; t < j; t++)

cout << a[k][t] << " ";

}

}

void delArr(int \*\*a, int i) {

for (int k = 0; k < i; k++)

delete[] a[k];

delete a;

}  
  
**Вывод**

Все поставленные задачи были выполнены. При выполнении использовались массивы с динамическим выделением памяти, которые создавались с помощью генератора случайных чисел. Для генератора случайных чисел была подключена библиотека ctime. Исходный код программ также доступен на GitHub по ссылке:  
  
<https://github.com/WarerBit/OOP/tree/master/P4>

**Практическая работа №5**

**Перегрузка функций**

**Цель практической работы**

Целью данной практической работы является приобретение практических навыков по программированию перегрузки функций на языке C++.

**Задачи**

1. Реализовать сортировку пузырьком для целых чисел, а затем перегрузить её для дробных.
2. Реализовать сортировку выбором для целых чисел, а затем перегрузить её для дробных.
3. Реализовать сортировку вставками для целых чисел, а затем перегрузить её для дробных.
4. Реализовать программу-калькулятор, работающую с разными типами данных.

**Ход работы**

Программа 1  
#include <iostream>

using namespace std;

void bubSort(int \*a);//Сортировка пузырьком

void bubSort(double \*a);

int main()

{

int \*a = new int[10];

cout << "Vvedite 10 celih chisel:";

for (int i = 0; i < 10; i++)

cin >> a[i];

bubSort(a);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << a[i];

delete[] a;

double \*b = new double[10];

cout << "\nVvedite 10 drobnih chisel:";

for (int i = 0; i < 10; i++)

cin >> b[i];

bubSort(b);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << b[i];

delete[] b;

}

void bubSort(int \*a) {

int k = 1; int n = 9;

while (k != 0) {

k = 0;

for (int i = 0; i < n; i++)

if (a[i] <= a[i + 1]) {

swap(a[i], a[i + 1]);

k++;

}

if (k != 0)

n--;

}

}

void bubSort(double \*a) {

int k = 1; int n = 9;

while (k != 0) {

k = 0;

for (int i = 0; i < n; i++)

if (a[i] <= a[i + 1]) {

swap(a[i], a[i + 1]);

k++;

}

if (k != 0)

n--;

}

}

Программа 2  
#include <iostream>

using namespace std;

void choSort(int \*a);//Сортировка выбором

void choSort(double \*a);

int main()

{

int \*a = new int[10];

cout << "Vvedite 10 celih chisel:";

for (int i = 0; i < 10; i++)

cin >> a[i];

choSort(a);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << a[i];

delete[] a;

double \*b = new double[10];

cout << "\nVvedite 10 drobnih chisel:";

for (int i = 0; i < 10; i++)

cin >> b[i];

choSort(b);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << b[i];

delete[] b;

}

void choSort(int \*a) {

int n = 9; int max;

while (n - 1 != 0) {

max = 0;

for (int i = 0; i < n; i++)

if (a[max] <= a[i])

max = i;

swap(a[max], a[n]);

n--;

}

}

void choSort(double \*a) {

int n = 9; int max;

while (n - 1 != 0) {

max = 0;

for (int i = 0; i < n; i++)

if (a[max] <= a[i])

max = i;

swap(a[max], a[n]);

n--;

}

}

Программа 3  
  
#include <iostream>

using namespace std;

void insSort(int \*a);//Сортировка вставками

void insSort(double \*a);

int main()

{

int \*a = new int[10];

cout << "Vvedite 10 celih chisel:";

for (int i = 0; i < 10; i++)

cin >> a[i];

insSort(a);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << a[i];

delete[] a;

double \*b = new double[10];

cout << "\nVvedite 10 drobnih chisel:";

for (int i = 0; i < 10; i++)

cin >> b[i];

insSort(b);

cout << "\nOtsortirovanniy massiv:";

for (int i = 0; i < 10; i++)

cout << " " << b[i];

delete[] b;

}

void insSort(int \*a) {

for (int i = 0; i < 10; i++)

for (int j = i; j > 0; j--)

if (a[j] < a[j - 1])

swap(a[j], a[j - 1]);

}

void insSort(double \*a) {

for (int i = 0; i < 10; i++)

for (int j = i; j > 0; j--)

if (a[j] < a[j - 1])

swap(a[j], a[j - 1]);

}  
  
  
Программа 4  
  
#include <iostream>

using namespace std;

void sum(int a, int b);//Сложение

void sum(double a, double b);

void dif(int a, int b);//Вычитание

void dif(double a, double b);

void prod(int a, int b);//Умножение

void prod(double a, double b);

void quot(int a, int b);//Деление

void quot(double a, double b);

int main()

{

char z; int n, a, b; double a1, b1;

cout << "S kakimi chislami vi hotite rabotat'? 1 - celie, 0 - drobnie:";

cin >> n;

cout << "Vvedite 2 chisla:";

if (n)

cin >> a >> b;

else

cin >> a1 >> b1;

cout << "Vvedite znak(+,-,\*,/):";

cin >> z;

switch (z)

{

case '+':

{

if (n)

sum(a, b);

else

sum(a1, b1);

break;

}

case '-':

{

if (n)

dif(a, b);

else

dif(a1, b1);

break;

}

case '\*':

{

if (n)

prod(a, b);

else

prod(a1, b1);

break;

}

case '/':

{

if (n)

quot(a, b);

else

quot(a1, b1);

break;

}

default:

break;

}

system("pause");

}

void sum(int a, int b) {

int d = a + b;

cout << a << "+" << b << "=" << d << endl;

}

void sum(double a, double b) {

double d = a + b;

cout << a << "+" << b << "=" << d << endl;

}

void dif(int a, int b) {

int d = a - b;

cout << a << "-" << b << "=" << d << endl;

}

void dif(double a, double b) {

double d = a - b;

cout << a << "-" << b << "=" << d << endl;

}

void prod(int a, int b) {

int d = a \* b;

cout << a << "\*" << b << "=" << d << endl;

}

void prod(double a, double b) {

double d = a \* b;

cout << a << "\*" << b << "=" << d << endl;

}

void quot(int a, int b) {

double d = a / b;

cout << a << "/" << b << "=" << d << endl;

}

void quot(double a, double b) {

double d = a / b;

cout << a << "/" << b << "=" << d << endl;

}

**Вывод**

Все задачи были выполнены. В ходе выполнения была изучена техника перегрузки функций. Исходный код программ также доступен на GitHub по ссылке:  
<https://github.com/WarerBit/OOP/tree/master/P5>

**Практическая работа №6**  
**Решение задач на наследование на языке С++**

**Цель практической работы**

Целью данной практической работы является приобретение практических навыков использования принципа ООП – наследования для разработки программ на языке программирования C++.

**Задачи**

1. Написать иерархию классов, описывающих имущество налогоплательщиков. Она должна состоять из абстрактного базового класса Property и производных от него классов Appartment, Car и CountryHouse. Базовый класс должен иметь поле worth (стоимость), конструктор с одним параметром, заполняющий это поле, и чисто виртуальный метод рассчета налога, переопределенный в каждом из производных классов. Налог на квартиру вычисляется как 1/1000 ее стоимости, на машину – 1/200, на дачу – 1/500. Также, каждый производный класс должен иметь конструктор с одним параметром, передающий свой параметр конструктору базового класса. В функции main завести массив из 7 указателей на Property и заполнить его указателями на динамические объекты производных классов (первые 3 – Appartment, следующие 2 – Car и последние 2 – CountryHouse). Вывести на экран величину налога для всех 7 объектов. Не забудь также уничтожить динамические объекты перед завершением программы.
2. Написать набор классов, представляющий выражения. В этом наборе должен быть один абстрактный базовый тип, а также набор производных от него типов по видам выражений (константа, переменная, сумма, разность, произведение, частное, sin, cos, exp, ln). У каждого из классов должны быть следующие виртуальные функции: напечатать выражение (без параметров), вычислить выражение (параметр – значение переменной, результат – значение выражения), вернуть производную выражения (без параметров), создать копию выражения (тоже без параметров).

**Ход работы**

Программа 1  
Файл Property.h  
#pragma once

class Property//Базовый класс

{

protected:

double worth;

public:

Property(double worth) {

this->worth = worth;

}

virtual double tax() = 0;//Виртуальный метод

virtual ~Property() {}

};

class Appartment :public Property//Производный класс

{

public:

Appartment(double worth) : Property(worth) {};

double tax() {

return worth / 1000;

}

};

class Car :public Property//Производный класс

{

public:

Car(double worth) : Property(worth) {};

double tax() {

return worth / 200;

}

};

class CountryHouse :public Property//Производный класс

{

public:

CountryHouse(double worth) : Property(worth) {};

double tax() {

return worth / 500;

}

};  
  
Файл Source.cpp

#include <iostream>

#include "Property.h"

using namespace std;

int main()

{

Property \*a[7];

for (int i = 0; i < 7; i++)

{

double worth;

cout << "Vvedite stoimost':";

cin >> worth;

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

if (i < 3)

a[i] = new Appartment(worth);

if (i < 5)

a[i] = new Car(worth);

else

a[i] = new CountryHouse(worth);

}

cout << "Taxes:" << endl;

for (int i = 0; i < 7; i++)

cout << i + 1 << ". " << a[i]->tax() << endl;

system("pause");

delete[] a;

return 0;

}  
  
Программа 2  
Файл Function.h  
#pragma once

#include <iostream>

#include <cmath>

using namespace std;

class Function//Базовый класс

{

public:

virtual void printF() = 0;//Виртуальные методы

virtual double count(double x) = 0;

virtual void takeP() = 0;

};

class Const : public Function {//Производный класс

double x;

public:

Const(double x) { this->x = x; }

void printF() {

cout << "f(x) = " << x << endl;

}

double count(double y) {

return x;

}

void takeP() {

cout << "f'(x) = 0" << endl;

}

};

class Argument : public Function {//Производный класс

public:

void printF() {

cout << "f(x) = x" << endl;

}

double count(double x) {

return x;

}

void takeP() {

cout << "f'(x) = 1" << endl;

}

};

class Sum : public Function {//Производный класс

double x1;

public:

Sum(double x1) { this->x1 = x1; }

void printF() {

cout << "f(x) = x + y" << endl;

}

double count(double x) {

return x + x1;

}

void takeP() {

cout << "f'(x) = 1" << endl;

}

};

class Dif : public Function {//Производный класс

double x1;

public:

Dif(double x1) { this->x1 = x1; }

void printF() {

cout << "f(x) = x - y" << endl;

}

double count(double x) {

return x - x1;

}

void takeP() {

cout << "f'(x) = 1 " << endl;

}

};

class Prod : public Function {//Производный Класс

double x1;

public:

Prod(double x1) { this->x1 = x1; }

void printF() {

cout << "f(x) = x \* y" << endl;

}

double count(double x) {

return x \* x1;

}

void takeP() {

cout << "f'(x) = y " << endl;

}

};

class Quot : public Function {//Производный класс

double x1;

public:

Quot(double x1) { this->x1 = x1; }

void printF() {

cout << "f(x) = x/y" << endl;

}

double count(double x) {

return x / x1;

}

void takeP() {

cout << "f'(x) = 1/y" << endl;

}

};

class Sin : public Function {//Производный класс

public:

void printF() {

cout << "f(x) = sin(x)" << endl;

}

double count(double x) {

return sin(x);

}

void takeP() {

cout << "f'(x) = cos(x)" << endl;

}

};

class Cos : public Function {//Производный класс

public:

void printF() {

cout << "f(x) = cos(x)" << endl;

}

double count(double x) {

return cos(x);

}

void takeP() {

cout << "f'(x) = -sin(x)" << endl;

}

};

class Exp : public Function {//Производный класс

public:

void printF() {

cout << "f(x) = e^x" << endl;

}

double count(double x) {

return exp(x);

}

void takeP() {

cout << "f'(x) = e^x" << endl;

}

};

class Ln : public Function {//Производный класс

public:

void printF() {

cout << "f(x) = ln(x)" << endl;

}

double count(double x) {

return log(x);

}

void takeP() {

cout << "f'(x) = 1/x" << endl;

}

};

Файл Source.cpp  
#include "Function.h"

#include <string>

int main()

{

double x, y; int ch;

cout << "Choose:\n 0 - Const\n 1 - Argument\n 2 - Sum\n 3 - Difference\n 4 - Product\n 5 - Quotient\n 6 - Sin\n 7 - Cos\n 8 - Exp\n 9 - Ln" << endl << "Entered:";

cin >> ch;

switch (ch) {

case 0: {

cout << "Enter const:";

cin >> y;

cout << "Enter x:";

cin >> x;

Const c(y);

c.printF();

cout << "f(" << x << ") = " << c.count(x) << endl;

c.takeP();

break;

}

case 1: {

cout << "Enter x:";

cin >> x;

Argument a;

a.printF();

cout << "f(" << x << ") = " << a.count(x) << endl;

a.takeP();

break;

}

case 2: {

cout << "Enter x , y:";

cin >> x >> y;

Sum sum(y);

sum.printF();

cout << "f(" << x << ") = " << sum.count(x) << endl;

sum.takeP();

break;

}

case 3: {

cout << "Enter x , y:";

cin >> x >> y;

Dif dif(y);

dif.printF();

cout << "f(" << x << ") = " << dif.count(x) << endl;

dif.takeP();

break;

}

case 4: {

cout << "Enter x , y:";

cin >> x >> y;

Prod prod(y);

prod.printF();

cout << "f(" << x << ") = " << prod.count(x) << endl;

prod.takeP();

break;

}

case 5: {

cout << "Enter x , y:";

cin >> x >> y;

Quot quot(y);

quot.printF();

cout << "f(" << x << ") = " << quot.count(x) << endl;

quot.takeP();

break;

}

case 6: {

cout << "Enter x:";

cin >> x;

Sin s;

s.printF();

cout << "f(" << x << ") = " << s.count(x) << endl;

s.takeP();

break;

}

case 7: {

cout << "Enter x:";

cin >> x;

Cos co;

co.printF();

cout << "f(" << x << ") = " << co.count(x) << endl;

co.takeP();

break;

}

case 8: {

cout << "Enter x:";

cin >> x;

Exp ex;

ex.printF();

cout << "f(" << x << ") = " << ex.count(x) << endl;

ex.takeP();

break;

}

case 9: {

cout << "Enter x:";

cin >> x;

Ln L;

L.printF();

cout << "f(" << x << ") = " << L.count(x) << endl;

L.takeP();

break;

}

default:

break;

}

system("pause");

}

**Вывод**

Все поставленные задачи были выполнены. В ходе выполнения работы был изучен механизм наследования классов в C++. Также для выполнения работы были использованы абстрактные классы и чисто виртуальные методы базовых абстрактных классов. Исходный код программ также доступен на GitHub по ссылке:

<https://github.com/WarerBit/OOP/tree/master/P6>

**Практическая работа №7**  
**Абстрактные классы. Множественное наследование**

**Цель практической работы**

Целью данной практической работы является знакомство с абстрактными классами и механизмом виртуальных функций, а так же множественным наследованием на языке программирования C++.

**Задачи**

1. Реализовать абстрактный класс «Животное» и путём наследования от него получить классы «Кошка», «Собака», «Попугай».
2. Реализовать абстрактный класс «Фигура» и путём наследования от него получить абстрактный класс «Четырёхугольник», и затем путём наследования получить классы «Ромб», «Прямоугольник».
3. Реализовать абстрактный класс «Транспортное средство» и путём наследования от него получить классы «Автомобиль», «Автобус», «Велосипед».
4. Реализовать абстрактные классы «Экран» и «Клавиатура», путём наследования от них получить классы «Ноутбук», «Телефон», «Стационарный компьютер».

**Ход работы**

Программа 1  
Файл Animal.h

#pragma once

#include <string>

#include <iostream>

using namespace std;

class Animal//Базовый класс

{

protected:

int age;

int legs;

string color;

public:

Animal(int, int, string);

virtual void sound() = 0;//Виртуальный метод

void setLegs(int legs) {

this->legs = legs;

}

void setAge(int age) {

this->age = age;

}

void setColor(string color) {

this->color = color;

}

int getLegs() {

return legs;

}

int getAge() {

return age;

}

string getColor() {

return color;

}

};

class Cat : public Animal {//Производный класс

public:

Cat(int age, int legs, string color) : Animal(age, legs, color) {}

void sound() {

cout << "Nya~" << endl;

}

};

class Dog : public Animal {//Производный класс

public:

Dog(int age, int legs, string color) : Animal(age, legs, color) {}

void sound() {

cout << "Woof" << endl;

}

};

class Parrot : public Animal {//Производный класс

public:

Parrot(int age, int legs, string color) : Animal(age, legs, color) {}

void sound() {

cout << "I am ur father,Luke!" << endl;

}

};  
  
Файл Animal.cpp

#include "Animal.h"

Animal::Animal(int age, int legs, string color)

{

this->age = age;

this->legs = legs;

this->color = color;

}

Файл Source.cpp  
#include "Animal.h"

#include <time.h>

int main()

{

srand(time(NULL));

string sc = "black", sd = "white", sp = "rainbow";

int ch;

cout << "Enter:\n 1 - Cat\n 2 - Dog\n 3 - Parrot" << endl;

cin >> ch;

switch (ch)

{

case 1: {

Cat cat(rand() % 9 + 1, 4, sc);

cout << "Cat's age is " << cat.getAge() << ", it has " << cat.getLegs() << " legs,and it's color is " << sc << endl;

cout << "Cat says - ";

cat.sound();

break;

}

case 2: {

Dog dog(rand() % 9 + 1, 4, sd);

cout << "Dog's age is " << dog.getAge() << ", it has " << dog.getLegs() << " legs,and it's color is " << sd << endl;

cout << "Dog says - ";

dog.sound();

break;

}

case 3: {

Parrot parrot(rand() % 9 + 1, 2, sp);

cout << "Parrot's age is " << parrot.getAge() << ", it has " << parrot.getLegs() << " legs,and it's color is " << sp << endl;

cout << "Parrot says - ";

parrot.sound();

break;

}

default:

break;

}

system("pause");

}

Программа 2  
Файл Shape.h  
#pragma once

#include<iostream>

using namespace std;

class Shape//Базовый класс

{

protected:

int side1;

int side2;

public:

Shape(int, int);

virtual void S() = 0;//Виртуальный метод

void setSide1(int side) {

this->side1 = side;

}

void setSide2(int side) {

this->side2 = side;

}

int getSide1() {

return side1;

}

int getSide2() {

return side2;

}

};

class Quadra : public Shape {//Производный класс

public:

Quadra(int side1, int side2) : Shape(side1, side2) {}

virtual void S() = 0;

};

class Rect : public Quadra {//Производный класс

public:

Rect(int side1, int side2) : Quadra(side1, side2) {}

void S() {

cout << "S = a\*b = " << side1 << "\*" << side2 << " = " << side1 \* side2 << endl;

}

};

class Rhomb : public Quadra {//Производный класс

public:

Rhomb(int side1, int side2) : Quadra(side1, side2) {}

void S() {

cout << "S = (a\*b)/2 = " << "(" << side1 << "\*" << side2 << ")/2 = " << (side1\*side2) / 2 << endl;

}

};

Файл Shape.cpp  
#include "Shape.h"

Shape::Shape(int side1, int side2)

{

this->side1 = side1;

this->side2 = side2;

}  
  
Файл Source.cpp  
#include "Shape.h"

int main()

{

int a, b;

cout << "Rectangle:\nEnter length and width:";

cin >> a >> b;

Rect rect(a, b);

cout << "length = " << rect.getSide1() << "\nwidth = " << rect.getSide2() << endl;

rect.S();

cout << "Rhomb:\nEnter 2 diagonals:";

cin >> a >> b;

Rhomb rhomb(a, b);

cout << "1st diagonal = " << rhomb.getSide1() << "\n2nd diagonal = " << rhomb.getSide2() << endl;

rhomb.S();

system("pause");

}

Программа 3  
Файл Vehicle.h  
#pragma once

#include<string>

#include<iostream>

using namespace std;

class Vehicle//Базовый класс

{

protected:

int wheels;

int seats;

string color;

public:

Vehicle(int, int, string);

virtual void speed() = 0;//Виртуальный метод

void setWheels(int wheels) {

this->wheels = wheels;

}

void setSeats(int seats) {

this->seats = seats;

}

void setColor(string color) {

this->color = color;

}

int getWheels() {

return wheels;

}

int getSeats() {

return seats;

}

string getColor() {

return color;

}

};

class Bike : public Vehicle {//Производный класс

public:

Bike(int wheels, int seats, string color) : Vehicle(wheels, seats, color) {}

void speed() {

cout << "Bike is faster then legs.\n";

}

};

class Automobile : public Vehicle {//Производный класс

public:

Automobile(int wheels, int seats, string color) : Vehicle(wheels, seats, color) {}

void speed() {

cout << "Automobile is faster then bike or bus.\n";

}

};

class Bus : public Vehicle {//Производный класс

public:

Bus(int wheels, int seats, string color) : Vehicle(wheels, seats, color) {}

void speed() {

cout << "Bus is faster then bike.\n";

}

};

Файл Vehicle.cpp  
#include "Vehicle.h"

Vehicle::Vehicle(int wheels, int seats, string color)

{

this->wheels = wheels;

this->seats = seats;

this->color = color;

}

Файл Source.cpp  
#include"Vehicle.h"

#include<ctime>

int main()

{

srand(time(NULL));

int n; string color;

cout << "Choose:\n1 - Bike\n2 - Automobile\n3 - Bus\nEnter:";

cin >> n;

switch (n) {

case 1:

{

Bike bike(rand() % 3 + 2, rand() % 2 + 1, "0");

cout << "Enter bike's color:";

cin >> color;

bike.setColor(color);

cout << "Ur bike has " << bike.getWheels() << " wheels, " << bike.getSeats() << " seats, and it's color is " << bike.getColor() << endl;

bike.speed();

break;

}

case 2:

{

Automobile car(4, rand() % 4 \* 2 + 2, "0");

cout << "Enter car's color:";

cin >> color;

car.setColor(color);

cout << "Ur car has " << car.getWheels() << " wheels, " << car.getSeats() << " seats, and it's color is " << car.getColor() << endl;

car.speed();

break;

}

case 3:

{

Bus bus((rand() % 3) \* 2 + 4, rand() % 10 + 10, "0");

cout << "Enter bus's color:";

cin >> color;

bus.setColor(color);

cout << "Ur bus has " << bus.getWheels() << " wheels, " << bus.getSeats() << " seats, and it's color is " << bus.getColor() << endl;

bus.speed();

break;

}

default:

break;

}

system("pause");

}

Программа 4  
Файл KandS.h  
#pragma once

#include<iostream>

#include<ctime>

using namespace std;

class Keyboard//Базовый класс

{

protected:

int buttons;

public:

Keyboard(int buttons) {

this->buttons = buttons;

}

virtual void slam() = 0;//Виртуальный метод

void setButtons(int buttons) {

this->buttons = buttons;

}

int getButtons() {

return buttons;

}

};

class Screen//Базовый класс

{

protected:

int length;

int width;

public:

Screen(int length, int width) {

this->length = length;

this->width = width;

}

virtual void punch() = 0;//Виртуальный метод

void setSize(int length, int width) {

this->length = length;

this->width = width;

}

int getLength() {

return length;

}

int getWidth() {

return width;

}

};

class PC : public Screen, public Keyboard {//Производный класс

public:

PC(int buttons, int length, int width) : Keyboard(buttons), Screen(length, width) {}

void slam() {

srand(time(NULL));

this->buttons = rand() % 50 + 25;

cout << "U've slammed ur keyboard,so there are left only " << this->buttons << " buttons...\n";

}

void punch() {

cout << "U've punched ur screen,so there is no screen anymore...\n";

}

};

class PCP : public Screen, public Keyboard {//Производный класс

public:

PCP(int buttons, int length, int width) : Keyboard(buttons), Screen(length, width) {}

void slam() {

cout << "U've slammed ur PCP,so there is no PCP anymore...\n";

}

void punch() {

cout << "U've punched ur PCP's screen,so there is no PCP's screen anymore... But hey! U still has PCP's keyboard ;)\n";

}

};

class Phone : public Screen, public Keyboard {//Производный класс

public:

Phone(int buttons, int length, int width) : Keyboard(buttons), Screen(length, width) {}

void slam() {

cout << "U've slammed ur phone...\nWhy did u do that? It was pretty expensive phone...\n";

}

void punch() {

cout << "U've punched ur phone,so... oh,wait,it was nokia 3310,so I guess u've broken ur arm...\n";

}

};  
  
Файл Source.cpp  
#include "KandS.h"

int main()

{

srand(time(NULL));

int n;

cout << "Choose:\n1 - PC\n2 - PCP\n3 - Phone\nEnter:";

cin >> n;

switch (n) {

case 1: {

PC pc(100, rand() % 10 \* 10, rand() % 10 \* 10);

cout << "Ur PC's keyboard has " << pc.getButtons() << " buttons and the size of ur screen is " << pc.getLength() << "\*" << pc.getWidth() << endl;

cout << "Do u wanna slam(1) or punch(2) or nothing(0)?Enter:";

cin >> n;

if (n == 1)

pc.slam();

if (n == 2)

pc.punch();

break;

}

case 2: {

PCP pcp(100, rand() % 10 \* 10, rand() % 10 \* 10);

cout << "Ur PCP's keyboard has " << pcp.getButtons() << " buttons and the size of ur screen is " << pcp.getLength() << "\*" << pcp.getWidth() << endl;

cout << "Do u wanna slam(1) or punch(2) or nothing(0)?Enter:";

cin >> n;

if (n == 1)

pcp.slam();

if (n == 2)

pcp.punch();

break;

}

case 3: {

Phone phone(100, rand() % 10, rand() % 10);

cout << "Ur Phone has " << phone.getButtons() << " buttons and the size of ur screen is " << phone.getLength() << "\*" << phone.getWidth() << endl;

cout << "Do u wanna slam(1) or punch(2) or nothing(0)?Enter:";

cin >> n;

if (n == 1)

phone.slam();

if (n == 2)

phone.punch();

break;

}

default:

break;

}

system("pause");

}

**Вывод**

Все задачи были выполнены. В ходе выполнения работы были более плотно изучены множественное наследование и абстрактные классы. Для достижения поставленной цели была использована библиотека cmath. Исходный код программ также доступен на GitHub по ссылке:

<https://github.com/WarerBit/OOP/tree/master/P7>

**Практическая работа №8**

**Работа с файлами в языке С++**

**Цель практической работы**

Целью данной практической работы является приобретение практических навыков по работе с файлами на языке программирования языке C++.

**Задачи**

1. Реализуйте программу, считывающую текст из файла и выводящую каждое слово с новой строки.
2. Реализуйте программу, считывающую текст с клавиатуры и записывающую его в файл.

**Ход работы**

Программа 1  
#include <iostream>

#include<string>

#include<Windows.h>

#include <fstream>

using namespace std;

int main()

{

ifstream filex("dingdongpingpong.txt");//Открытие файла для чтения

string s;

while (!filex.eof()) {//Считывание текста из файла и вывод каждого слова с новой строки

filex >> s;

cout << s << endl;

}

filex.close();//Закрытие файла

system("pause");

}  
  
dingdongpingpong.txt:  
DING DONG PINGPONG

Программа 2  
#include <iostream>

#include<Windows.h>

#include <fstream>

using namespace std;

int main()

{

ofstream filex("dingdongpingpong.txt");//Открытие файла для записи

char s;

cin.get(s);

while (s != '\n') {//Запись в файл

filex << s;

cin.get(s);

}

filex.close();//Закрытие файла

system("pause");

}

**Вывод**

Все задачи были выполнены успешно. В ходе работы была исследована работа с файлами. Для достижения результата использовалась библиотек fstream, позволяющая производить работу с файлами. Исходный код программ также доступен на GitHub по ссылке:  
  
<https://github.com/WarerBit/OOP/tree/master/P8>