|  |  |
| --- | --- |
| 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 | Рындин Д.П. |
| Принял ассистент кафедры | Хлебникова В.Л. |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «23» февраля 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**

1. Реализовать программу, которая считывает имя и здоровается

**#include** <iostream>

**#include** <string>

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

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

**setlocale**(LC\_ALL, "");

string name;

cout << "Enter your name: ";

cin >> name;

cout << "Hello, " << name << "!" << **endl**;

**system**("pause");

**return** 0;

}

1. Реализовать программу, производящую операции над числами. (Сложение/вычитание, целые числа/числа с плавающей запятой)

**#include** <iostream>

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

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

**setlocale**(LC\_ALL, "");

**double** a, b;

cout << "Enter two numbers: ";

cin >> a >> b;

cout << "a - b = " << (a - b) << **endl**;

cout << "a + b = " << (a + b) << **endl**;

**system**("pause");

**return** 0;

}

**Вывод**

В ходе выполнения практической работы была использована библиотека стандартного ввода и вывода, операторы ввода, вывода, сложения и вычитания. Были достигнуты результаты, изображённые на скриншотах (рис. 1 и 2). Также исходный код программ был помещён на GitHub и доступен по ссылкам:

1. <https://github.com/dimaINBO0418/oop/blob/master/1.%D0%98%D0%BC%D1%8F.txt>

<https://github.com/dimaINBO0418/oop/blob/master/1.%D0%9A%D0%B0%D0%BB%D1%8C%D0%BA%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80.txt>

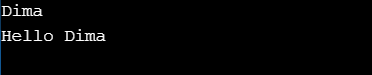


Рис. 1 Результат работы первой программы

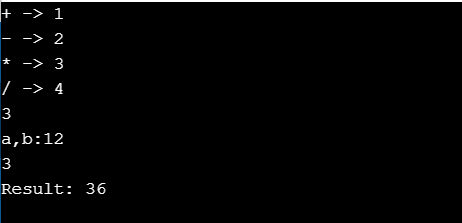


Рис. 2 Результат работы второй программы

## Практическая работа №2

**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**

1. Вариант 1

**#include** <iostream>

**#include** <cmath>

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

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

**double** a, b, m, n;

cout << "Enter a and b: ";

cin >> a >> b;

**if** (a >= b) {

n = **cbrt**(a - b);

} **else** {

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

}

**if** (n < b) {

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

} **else** {

**if** (n == b) {

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

} **else** {

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

}

}

**if** (isnan(m) || isnan(n) || isinf(m) || isinf(n)) {

cout << "Error: cannot count result" << **endl**;

} **else** {

cout << "a = " << a << **endl**;

cout << "b = " << b << **endl**;

cout << "m = " << m << **endl**;

cout << "n = " << n << **endl**;

}

**system**("pause");

**return** 0;

}

1. Вариант 2

**#include** <iostream>

**#include** <cmath>

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

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

**double** a, b, z, t;

cout << "Enter a and b: ";

cin >> a >> b;

**if** (a < b) {

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

} **else** {

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

}

**if** (z < b) {

t = **cbrt**(z + a \* a \* b);

} **else** {

**if** (z == b) {

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

} **else** {

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

}

}

**if** (isnan(t) || isnan(z) || isinf(z) || isinf(t)) {

cout << "Error: cannot count result" << **endl**;

} **else** {

cout << "a = " << a << **endl**;

cout << "b = " << b << **endl**;

cout << "t = " << t << **endl**;

cout << "z = " << z << **endl**;

}

**system**("pause");

**return** 0;

}

1. Вариант 3

**#include** <iostream>

**#include** <cmath>

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

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

**double** a, b, y, t;

cout << "Enter a and b: ";

cin >> a >> b;

**if** (a <= b) {

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

} **else** {

y = a + pow(**log**(b), 2);

}

**if** (y == b) {

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** {

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

}

}

**if** (isnan(y) || isnan(t) || isinf(y) || isinf(t)) {

cout << "Error: cannot count result" << **endl**;

} **else** {

cout << "a = " << a << **endl**;

cout << "b = " << b << **endl**;

cout << "y = " << y << **endl**;

cout << "t = " << t << **endl**;

}

**system**("pause");

**return** 0;

}

1. Вариант 4

**#include** <iostream>

**#include** <cmath>

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

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

**double** a, x, y, t;

cout << "Enter a and x: ";

cin >> a >> x;

**if** (a <= x) {

y = a + **log**(x + a);

} **else** {

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

}

**if** (a > y) {

t = y / (a - x);

} **else** {

**if** (a == y) {

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

} **else** {

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

}

}

**if** (isnan(y) || isnan(t) || isinf(y) || isinf(t)) {

cout << "Error: cannot count result" << **endl**;

} **else** {

cout << "a = " << a << **endl**;

cout << "x = " << x << **endl**;

cout << "y = " << y << **endl**;

cout << "t = " << t << **endl**;

}

**system**("pause");

**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». Были достигнуты результаты, изображённые на скриншотах (рис. 3-6). Также код программ доступен по ссылкам:

1. <https://github.com/dimaINBO0418/oop/blob/master/2.1.txt>
2. <https://github.com/dimaINBO0418/oop/blob/master/2.2.txt>
3. <https://github.com/dimaINBO0418/oop/blob/master/2.3.txt>
4. <https://github.com/dimaINBO0418/oop/blob/master/2.4.txt>

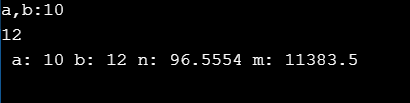


Рис. 3 Результат практической работы №2 вариант 1

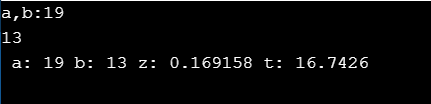


Рис. 4 Результат практической работы №2 вариант 2

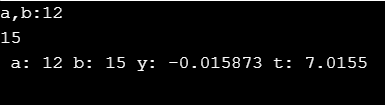


Рис. 5 Результат практической работы №2 вариант 3

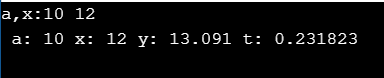


Рис. 6 Результат практической работы №2 вариант 4

## 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**

1. Реализовать класс «Книга»

|  |
| --- |
|  |
|  | #include <iostream>  #include <string> |
|  | using namespace std; |
|  | class Kniga |
|  | { |
|  | public: |
|  | string autor; |
|  | string Name; |
|  | double stranica; |
|  | void open(); |
|  | }; |
|  | int main() |
|  | { |
|  | Kniga Book; |
|  | Book.open(); |
|  | system("pause"); |
|  | return 0; |
|  |  |
|  | } |
|  | void Kniga::open() |
|  | { |
|  | double random; |
|  | cout << "vvedite avtora" << endl; |
|  | cin >> autor; |
|  | cout << "vvedite nazvanie" << endl; |
|  | cin >> Name; |
|  | random = 1 + rand() % 50; |
|  | cout << " " << autor << " " << "" << Name << " " << " otkryta na" <<" "<< random <<" "<< "stranice" << endl; |
|  | } |

1. Реализовать класс «Группа студентов»

|  |
| --- |
|  |
|  | #include <iostream>  #include <string> |
|  | using namespace std; |
|  |  |
|  | class Students { |
|  | string \*members; |
|  | int n; |
|  | public: |
|  | void makestudents(); |
|  | void getstudent(); |
|  | }; |
|  | void Students::makestudents() { |
|  | cout << "vvedite kolichestvo studentov" << endl; |
|  | cin >> n; |
|  | members = new string[n]; |
|  | for (int i = 0; i < n; i++){cout<<"vvedite familiyu studenta"<<endl; |
|  | cin >> members[i]; } |
|  |  |
|  | } |
|  | void Students::getstudent() { |
|  | for (int i = 0; i < n; i++)cout<<"student "<<i+1<<" " << members[i]<< endl; |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | Students a; |
|  | a.makestudents(); |
|  | a.getstudent(); |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Реализовать класс «Автомобиль»

|  |
| --- |
|  |
|  | #include <iostream>  #include <string> |
|  |  |
|  | using namespace std; |
|  |  |
|  | class avto{ |
|  | int skorost; |
|  | int god; |
|  | string nazvanie; |
|  |  |
|  | public: |
|  | avto() { |
|  | god = 0; |
|  | skorost = 0; |
|  | nazvanie = "net nazvaniya"; |
|  | } |
|  | void set\_avto() { |
|  | cout<<"vvedite god vypuska"<<endl; |
|  | cin >> this->god; |
|  | cout<<"vvedite maksimalnuyu skorost"<<endl; |
|  | cin>>this->skorost ; |
|  | cout<<"vvedite nazvaniye"<<endl; |
|  | cin>>this->nazvanie ; |
|  |  |
|  | } |
|  | void get\_avto() { |
|  | cout << god<<endl; |
|  | cout << skorost <<"km/h"<< endl; |
|  | cout << nazvanie << endl; |
|  |  |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | avto a; |
|  | a.set\_avto(); |
|  | a.get\_avto(); |
|  |  |
|  |  |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Реализовать класс «Вектор»

|  |
| --- |
|  |
|  | #include <iostream>  #include <string> |
|  | #include <cmath> |
|  | using namespace std; |
|  | class vector { |
|  | int x, y; |
|  |  |
|  | public: |
|  | vector(){ |
|  | x = 0, y = 0; |
|  |  |
|  | } |
|  | vector(int x, int y) { |
|  | this->x = x; |
|  | this->y = y; |
|  | } |
|  | void set\_vector() { |
|  | int x1, y1, x2, y2; |
|  | cout << "vvedite koordinati tochec" << endl; |
|  | cout << "x1, y1"; |
|  | cin >> x1 >> y1; |
|  | cout<<endl; |
|  | cout << "x2, y2"; |
|  | cin >> x2 >> y2; |
|  | cout << endl; |
|  | x = x2 - x1; |
|  | y = y2 - y1; |
|  |  |
|  | } |
|  |  |
|  | double modul() { |
|  | return sqrt(x\*x + y\*y); |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | vector a; |
|  | vector b(3,4); |
|  | a.set\_vector(); |
|  | cout <<" modul vectora raven "<< a.modul() << endl;; |
|  |  |
|  |  |
|  |  |
|  |  |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Реализовать класс «Библиотека»

|  |
| --- |
|  |
|  |
|  | #include <iostream>  #include <string> |
|  | #include <cmath> |
|  | using namespace std; |
|  | class vector { |
|  | int x, y; |
|  |  |
|  | public: |
|  | vector(){ |
|  | x = 0, y = 0; |
|  |  |
|  | } |
|  | vector(int x, int y) { |
|  | this->x = x; |
|  | this->y = y; |
|  | } |
|  | void set\_vector() { |
|  | int x1, y1, x2, y2; |
|  | cout << "vvedite koordinati tochec" << endl; |
|  | cout << "x1, y1"; |
|  | cin >> x1 >> y1; |
|  | cout<<endl; |
|  | cout << "x2, y2"; |
|  | cin >> x2 >> y2; |
|  | cout << endl; |
|  | x = x2 - x1; |
|  | y = y2 - y1; |
|  |  |
|  | } |
|  |  |
|  | double modul() { |
|  | return sqrt(x\*x + y\*y); |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | vector a; |
|  | vector b(3,4); |
|  | a.set\_vector(); |
|  | cout <<" modul vectora raven "<< a.modul() << endl;; |
|  |  |
|  |  |
|  |  |
|  |  |
|  | system("pause"); |
|  | return 0; |
|  | } |

**Вывод**

Все поставленные задачи были выполнены. Для этого были использованы библиотека STL, перегрузка оператора равенства, шаблоны функций. Все задачи были выполнены в виде много файлового проекта с разбиением классов на файлы объявления и реализации класса. Результаты работы программ приведены в скриншотах (рис. 7-11). Код программ также доступен на GitHub по ссылкам:

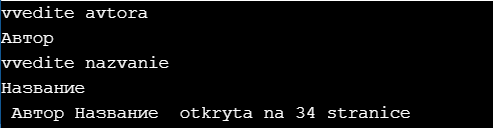
1. [<https://github.com/dimaINBO0418/oop/blob/master/3.1.txt>](https://github.com/VladRubel/laby/blob/master/3.1)
2. [[[https://github.com/dimaINBO0418/oop/blob/master/3.2.txt](https://github.com/VladRubel/laby/blob/master/3.2)](https://github.com/dimaINBO0418/oop/blob/master/3.1.txt)](https://github.com/VladRubel/laby/blob/master/3.1)
3. [[https://github.com/dimaINBO0418/oop/blob/master/3.3.txt](https://github.com/VladRubel/laby/blob/master/3.3)](https://github.com/VladRubel/laby/blob/master/3.1)
4. [[https://github.com/dimaINBO0418/oop/blob/master/3.4.txt](https://github.com/VladRubel/laby/blob/master/3.4)](https://github.com/VladRubel/laby/blob/master/3.1)
5. [[[https://github.com/dimaINBO0418/oop/blob/master/3.5.txt](https://github.com/VladRubel/laby/blob/master/3.5)](https://github.com/dimaINBO0418/oop/blob/master/3.1.txt)](https://github.com/VladRubel/laby/blob/master/3.1)
6. 

Рис. 7 Результат работы программы №1

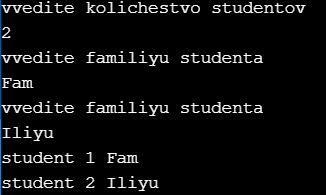


Рис. 8 Результат работы программы №2

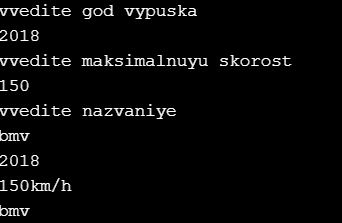


Рис. 9 Результат работы программы №3

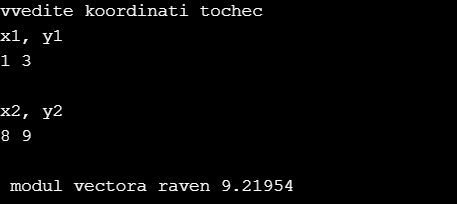


Рис 10 Результат работы программы №4

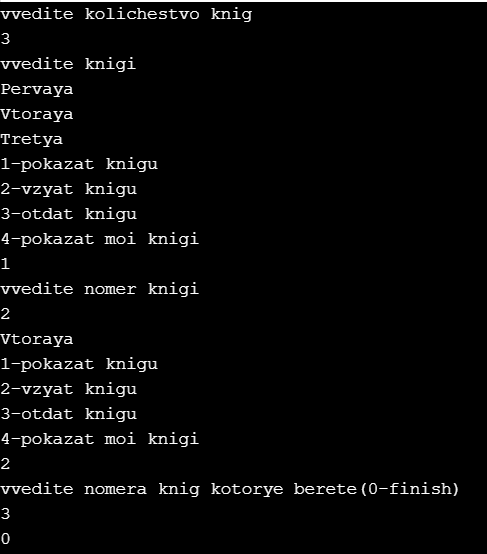


Рис. 11 Результат работы программы №5

## 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**

1. Программа №1

**#include** <iostream>

**#include** <ctime>

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

**double** \***createArray**(**int**); // Создаёт массив

**void** **fillArray**(**double** \*, **int**); // Заполняет массив случайными числами

**void** **printArray**(**double** \*, **int**); // Выводит массив на экран

**void** **deleteArray**(**double** \*, **int**); // Удаляет массив

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

**srand**(**time**(NULL));

**int** size;

cout << "Enter size of massive(lower than 1 if you want to exit the program): ";

cin >> size;

**while** (size > 0) {

**double** \*arr = createArray(size);

fillArray(arr, size);

printArray(arr, size);

deleteArray(arr, size);

cout << "Enter size of massive(lower than 1 if you want to exit the program): ";

cin >> size;

}

**return** 0;

}

**double** \***createArray**(**int** size) { **return** **new** **double**[size]; }

**void** **fillArray**(**double** \*arr, **int** size) {

**for** (**int** i = 0; i < size; i++) {

arr[i] = **rand**() % 100;

}

}

**void** **printArray**(**double** \*arr, **int** size) {

cout << "Array: ";

**for** (**int** i = 0; i < size; i++) {

cout << arr[i] << ' ';

}

cout << **endl**;

}

**void** **deleteArray**(**double** \*arr, **int** size) { **delete** [] arr; }

1. Программа №2

**#include** <iostream>

**#include** <ctime>

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

**void** **fillArray**(**int** \*, **int**); // Заполняет массив случайными числами

**void** **printArray**(**int** \*, **int**); // Выводит массив на экран

**void** **changeElements**(**int** \*, **int**); // Меняет местами пары элементов массива

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

**srand**(**time**(NULL));

**int** size = 12;

**int** \*arr = **new** **int**[size];

fillArray(arr, size);

printArray(arr, size);

changeElements(arr, 12);

printArray(arr, size);

**delete** [] arr;

**system**("pause");

**return** 0;

}

**void** **fillArray**(**int** \*arr, **int** size) {

**for** (**int** i = 0; i < size; i++) {

arr[i] = **rand**() % 100;

}

}

**void** **printArray**(**int** \*arr, **int** size) {

cout << "Array: ";

**for** (**int** i = 0; i < size; i++) {

cout << arr[i] << ' ';

}

cout << **endl**;

}

**void** **changeElements**(**int** \*arr, **int** size) {

**for** (**int** i = 0; i < size; i = i + 2) {

**int** c = arr[i];

arr[i] = arr[i+1];

arr[i+1] = c;

}

}

1. Программа №3

**#include** <iostream>

**#include** <ctime>

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

**void** **fillArray**(**int** \*\*, **int**, **int**); // Заполняет массив случайными числами

**void** **printArray**(**int** \*\*, **int**, **int**); // Выводит массив на экран

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

**srand**(**time**(NULL));

**int** m, n;

cout << "Enter number of rows and columns: ";

cin >> n >> m;

**int** \*\*arr = **new** **int**\*[n];

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

arr[i] = **new** **int**[m];

}

fillArray(arr, n, m);

printArray(arr, n, m);

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

**delete** [] arr[i];

}

**delete** [] arr;

**system**("pause");

**return** 0;

}

**void** **fillArray**(**int** \*\*arr, **int** n, **int** m) {

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

**for** (**int** j = 0; j < m; j++) {

arr[i][j] = **rand**() % 41 + 10;

}

}

}

**void** **printArray**(**int** \*\*arr, **int** n, **int** m) {

cout << "Array:" << **endl**;

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

**for** (**int** j = 0; j < m; j++) {

cout << arr[i][j] << ' ';

}

cout << **endl**;

}

}

**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. Результат работы программ приведён на скриншотах (рис.12-14). Так же исходный код программ доступен на GitHub по ссылкам:

1. [<https://github.com/dimaINBO0418/oop/blob/master/4.1.txt>](https://github.com/IvLaptev/OOP/tree/master/Pr4.1)
2. [https://github.com/dimaINBO0418/oop/blob/master/4.2.txt](https://github.com/IvLaptev/OOP/tree/master/Pr4.2)
3. [[https://github.com/dimaINBO0418/oop/blob/master/4.3.txt](https://github.com/dimaINBO0418/oop/blob/master/4.1.txt)](https://github.com/IvLaptev/OOP/tree/master/Pr4.3)

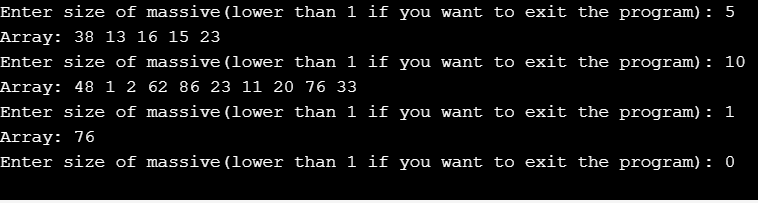


Рис. 12 Результат работы программы №1



Рис. 13 Результат работы программы №2

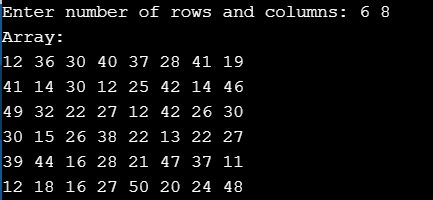


Рис. 14 Результат работы программы №3

## 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**

1. Программа №1

|  |
| --- |
|  |
|  | #include <iostream>  #include<ctime> |
|  | #include <iomanip> |
|  | using namespace std; |
|  | void sort(int \*a , int n ); |
|  | void sort(double \*a , int n ); |
|  | int main() { |
|  | srand(time(0)); |
|  | int n,z; |
|  | cout<<"vvedite kolichestvo chisel"<<endl; |
|  | cin >> n; |
|  | cout<<"kakiye chisla budete vvodit(0-double/1-int)"<<endl; |
|  | cin>>z; |
|  | if (z == 0) { |
|  | double\*a = new double[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | cout<<" vvedite chislo"<<endl; |
|  | cin >> a[i]; |
|  | } |
|  | sort(a, n); |
|  |  |
|  | for (int i = 0; i < n; i++) |
|  | cout << a[i] << endl; |
|  | } |
|  | else { |
|  | int\*a = new int[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | cout<<" vvedite chislo"<<endl; |
|  | cin >> a[i]; |
|  | } |
|  | sort(a, n); |
|  |  |
|  | for (int i = 0; i < n; i++) |
|  | cout << a[i] << endl; |
|  | } |
|  | system("pause"); |
|  | return 0; |
|  |  |
|  | } |
|  |  |
|  | void sort(double \*a, int n) |
|  | { |
|  | int i, j, tmp; |
|  | cout<<"idet sortirovka drobnyh chisel"<<endl; |
|  | for (i = 0; i < n; i++) |
|  | for (j = n - 1; j >= 0; j--) |
|  | if (a[j] < a[j + 1]) |
|  | { |
|  | tmp = a[j]; |
|  | a[j] = a[j + 1]; |
|  | a[j + 1] = tmp; |
|  | } |
|  | } |
|  |  |
|  | void sort(int \*a, int n) |
|  | { |
|  | int i, j, tmp; |
|  | cout<<"idet sortirovka celyh chisel"<<endl; |
|  | for (i = 0; i < n; i++) |
|  | for (j = n - 1; j >= 0; j--) |
|  | if (a[j] < a[j + 1]) |
|  | { |
|  | tmp = a[j]; |
|  | a[j] = a[j + 1]; |
|  | a[j + 1] = tmp; |
|  | } |
|  | } |

1. Программа №2

|  |
| --- |
|  |
|  | #include <iostream>  #include<ctime> |
|  | #include <iomanip> |
|  | using namespace std; |
|  | void sort(int \*a , int n ); |
|  | void sort(double \*a , int n ); |
|  | int main() { |
|  | srand(time(0)); |
|  | int n,z; |
|  | cout<<"vvedite kolichestvo chisel"<<endl; |
|  | cin >> n; |
|  | cout<<"kakiye chisla budete vvodit(0-double/1-int)"<<endl; |
|  | cin>>z; |
|  | if (z == 0) { |
|  | double\*a = new double[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | cout<<" vvedite chislo"<<endl; |
|  | cin >> a[i]; |
|  | } |
|  | sort(a, n); |
|  |  |
|  | for (int i = 0; i < n; i++) |
|  | cout << a[i] << endl; |
|  | } |
|  | else { |
|  | int\*a = new int[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | cout<<" vvedite chislo"<<endl; |
|  | cin >> a[i]; |
|  | } |
|  | sort(a, n); |
|  |  |
|  | for (int i = 0; i < n; i++) |
|  | cout << a[i] << endl; |
|  | } |
|  | system("pause"); |
|  | return 0; |
|  |  |
|  | } |
|  |  |
|  | void sort(double \*a, int n) |
|  | { |
|  | long i, j, k; |
|  | cout<<"idet sortirovka drobnyh chisel"<<endl; |
|  | double x; |
|  |  |
|  | for (i = 0; i < n; i++) { // i - номер текущего шага |
|  | k = i; x = a[i]; |
|  |  |
|  | for (j = i + 1; j < n; j++) // цикл выбора наименьшего элемента |
|  | if (a[j] < x) { |
|  | k = j; x = a[j]; // k - индекс наименьшего элемента |
|  | } |
|  |  |
|  | a[k] = a[i]; a[i] = x; // меняем местами наименьший с a[i] |
|  | } |
|  | } |
|  | void sort(int \*a, int n) |
|  | { |
|  | long i, j, k; |
|  | cout<<"idet sortirovka celyh chisel"<<endl; |
|  | double x; |
|  |  |
|  | for (i = 0; i < n; i++) { // i - номер текущего шага |
|  | k = i; x = a[i]; |
|  |  |
|  | for (j = i + 1; j < n; j++) // цикл выбора наименьшего элемента |
|  | if (a[j] < x) { |
|  | k = j; x = a[j]; // k - индекс наименьшего элемента |
|  | } |
|  |  |
|  | a[k] = a[i]; a[i] = x; // меняем местами наименьший с a[i] |
|  | } |
|  | } |

1. Программа №3

|  |
| --- |
|  |
|  | #include <iostream>  #include <ctime> |
|  | using namespace std; |
|  |  |
|  | void fillArray(int \*\*, int, int); // Заполняет массив случайными числами |
|  |  |
|  | void printArray(int \*\*, int, int); // Выводит массив на экран |
|  |  |
|  | int main() { |
|  | srand(time(NULL)); |
|  |  |
|  | int m, n; |
|  | cout << "Enter number of rows and columns: "; |
|  | cin >> n >> m; |
|  |  |
|  | int \*\*arr = new int\*[n]; |
|  | for (int i = 0; i < n; i++) { |
|  | arr[i] = new int[m]; |
|  | } |
|  |  |
|  | fillArray(arr, n, m); |
|  | printArray(arr, n, m); |
|  |  |
|  | for (int i = 0; i < n; i++) { |
|  | delete [] arr[i]; |
|  | } |
|  | delete [] arr; |
|  |  |
|  | system("pause"); |
|  | return 0; |
|  | } |
|  |  |
|  | void fillArray(int \*\*arr, int n, int m) { |
|  | for (int i = 0; i < n; i++) { |
|  | for (int j = 0; j < m; j++) { |
|  | arr[i][j] = rand() % 41 + 10; |
|  | } |
|  | } |
|  | } |
|  |  |
|  | void printArray(int \*\*arr, int n, int m) { |
|  | cout << "Array:" << endl; |
|  | for (int i = 0; i < n; i++) { |
|  | for (int j = 0; j < m; j++) { |
|  | cout << arr[i][j] << ' '; |
|  | } |
|  | cout << endl; |
|  | } |
|  | } |

1. Программа №4

|  |
| --- |
|  |
|  | #include<ctime> |
|  | #include <iomanip> |
|  | using namespace std; |
|  |  |
|  | double calc(int a, int b); |
|  | double calc(double a, double b); |
|  |  |
|  | int main() { |
|  | int z; |
|  | cout << "1-int,2-double" << endl; |
|  | cin >> z; |
|  | if (z == 1) { |
|  | int a , b ; |
|  | cout << "vvedite 2 chisla" << endl; |
|  | cin >> a >> b; |
|  | cout << calc(a, b) << endl; |
|  | } |
|  | else { |
|  | double a, b; |
|  | cout << "vvedite 2 chisla" << endl; |
|  | cin >> a >> b; |
|  | cout << calc(a, b) << endl; |
|  | } |
|  | system("pause"); |
|  | return 0; |
|  | } |
|  |  |
|  | double calc(int a, int b) |
|  | { |
|  | int k; |
|  | do { |
|  | cout << "i1+2-3\*4/" << endl; |
|  | cin >> k; |
|  | switch (k) |
|  | { |
|  | case 1: |
|  | return a + b; |
|  | break; |
|  | case 2: |
|  | return a - b; |
|  | break; |
|  | case 3: |
|  | return a\*b; |
|  | break; |
|  | case 4: |
|  | return a / b; |
|  | default: |
|  | break; |
|  | } |
|  |  |
|  | } while (k != 0); |
|  | } |
|  |  |
|  | double calc(double a, double b) |
|  | { |
|  | int k; |
|  | do { |
|  | cout << "d1+2-3\*4/" << endl; |
|  | cin >> k; |
|  | switch (k) |
|  | { |
|  | case 1: |
|  | return a + b; |
|  | break; |
|  | case 2: |
|  | return a - b; |
|  | break; |
|  | case 3: |
|  | return a\*b; |
|  | break; |
|  | case 4: |
|  | return a / b; |
|  | default: |
|  | break; |
|  | } |
|  |  |
|  | } while (k != 0); |
|  | } |

**Вывод**

Все задачи были выполнены. В ходе выполнения была изучена техника перегрузки функций. Так же для заполнения массивов использовался генератор случайных чисел. Результат работы программы приведён на скриншотах (рис. 15-18). Исходный код программ так же доступен на GitHub по ссылкам:

1. [https://github.com/dimaINBO0418/oop/blob/master/5.1.txt](https://github.com/VladRubel/laby/blob/master/5.1)
2. [https://github.com/dimaINBO0418/oop/blob/master/5.2.txt](https://github.com/VladRubel/laby/blob/master/5.2)
3. [https://github.com/dimaINBO0418/oop/blob/master/5.3.txt](https://github.com/VladRubel/laby/blob/master/5.3)
4. [https://github.com/dimaINBO0418/oop/blob/master/5.4.txt](https://github.com/VladRubel/laby/blob/master/5.4)

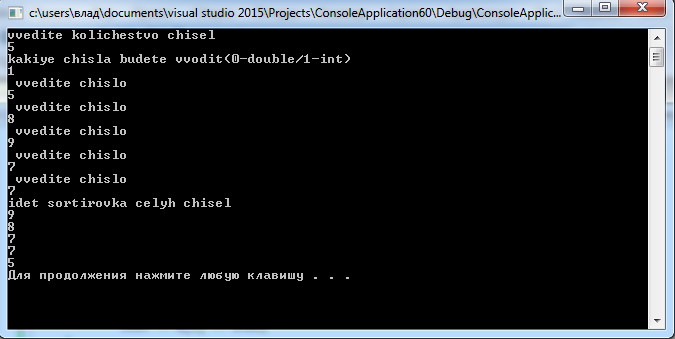


Рис. 15 Результат работы программы №1

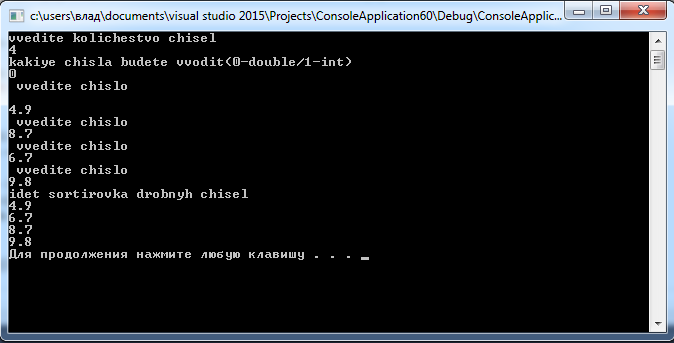


Рис. 16 Результат работы программы №2

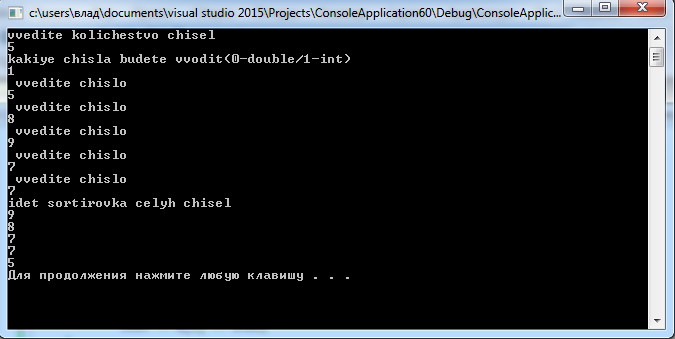


Рис. 17 Результат работы программы №3

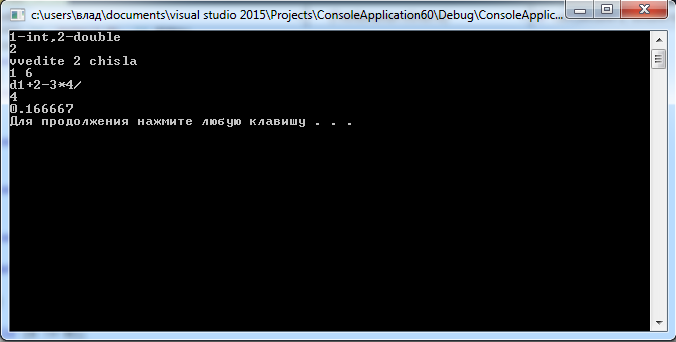


Рис. 18 Результат работы программы №4

## 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**

1. Программа №1

|  |
| --- |
|  |
|  | #include <iostream> |
|  | using namespace std; |
|  |  |
|  | class Property { |
|  | protected: |
|  | int worth; |
|  | public: |
|  | virtual double nalog() = 0; |
|  | Property(int worth) { |
|  | this->worth = worth; |
|  | } |
|  | virtual ~Property() {}; |
|  | }; |
|  |  |
|  | class Car: public Property |
|  | { |
|  | public: double nalog() { |
|  |  |
|  | return worth / 200; |
|  | } |
|  | Car(int worth) :Property(worth) { |
|  |  |
|  | } |
|  | }; |
|  |  |
|  | class CountryHouse : public Property |
|  | { |
|  | public: double nalog() { |
|  |  |
|  | return worth / 500; |
|  | } |
|  | CountryHouse(int worth) :Property(worth) { |
|  |  |
|  | } |
|  | }; |
|  |  |
|  | class Appartment : public Property |
|  | { |
|  | public: double nalog() { |
|  |  |
|  | return worth / 1000; |
|  | } |
|  | Appartment(int worth) :Property(worth) { |
|  |  |
|  | } |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | Property\* a[7]; |
|  | for (int i = 0; i < 7; i++) { |
|  | int worth; |
|  | cout << "vvedite stoimost ob'ecta " << i + 1 << endl; |
|  | cin >> worth; |
|  |  |
|  | if (i < 3)a[i] = new Appartment(worth); |
|  | else if (i < 5)a[i] = new Car(worth); |
|  | else a[i] = new CountryHouse(worth); |
|  | cout << "nalog" << i + 1 << a[i]->nalog() << endl; |
|  | } |
|  | delete[] \*a; |
|  | system("pause"); |
|  | return 0; |
|  | } |

Программа №2:

**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. Результаты работы программ представлены на скриншотах (рис 19, 20). Исходный код программ также доступен на GitHub по ссылкам:

1. [https://github.com/dimaINBO0418/oop/blob/master/6.1.txt](https://github.com/VladRubel/laby/blob/master/6.1)
2. [<https://github.com/dimaINBO0418/oop/blob/master/6.2.txt>](https://github.com/VladRubel/laby/blob/master/6.2)

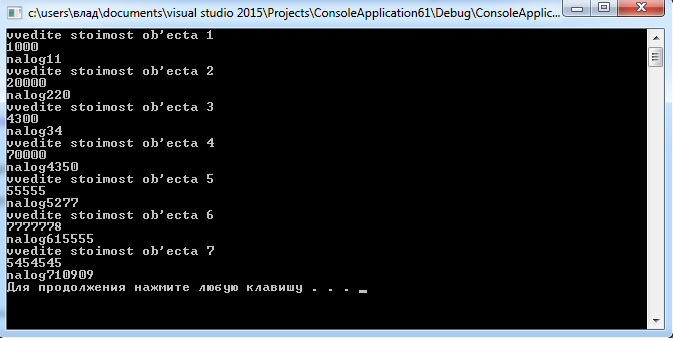


Рис. 19 Результат работы программы №1

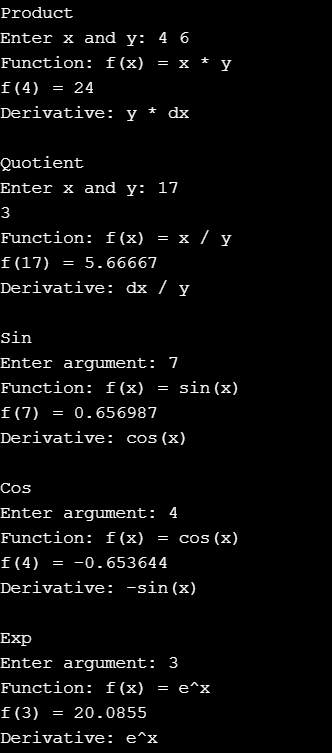
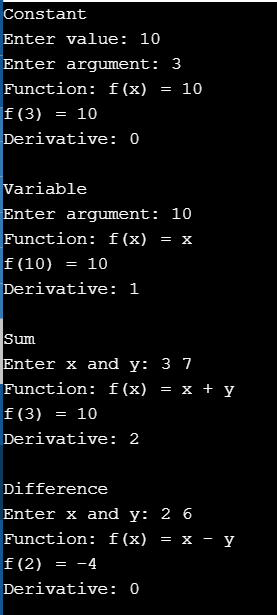


Рис. 20 Результат работы программы №2

## 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**

1. Программа №1

|  |
| --- |
|  |
|  | #include<string> |
|  |  |
|  | using namespace std; |
|  |  |
|  | class Animal { |
|  | protected: |
|  | string name; |
|  | int year; |
|  | public: |
|  | Animal(string name, int year) { |
|  | this->name = name; |
|  | this->year = year; |
|  | } |
|  | Animal() { |
|  | cout<<"vvedite imya i kolichestvo let"<<endl; |
|  | cin >> name; |
|  | cin >> year; |
|  | } |
|  | virtual void golos() = 0; |
|  | void get\_Data() { |
|  | cout << "Ya " << name << " " << year << endl; |
|  | } |
|  | }; |
|  |  |
|  | class Cat: public Animal { |
|  | public: |
|  | Cat(string name, int year):Animal(name,year){} |
|  | Cat():Animal(){} |
|  | void golos() { |
|  | cout << "myaaaaaau!!!"<<endl; |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | class Dog : public Animal { |
|  | public: |
|  | Dog(string name, int year) :Animal(name, year) {} |
|  | Dog() :Animal() {} |
|  | void golos() { |
|  | cout << "gav gav gav!!!" << endl; |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | class Parrot : public Animal { |
|  | public: |
|  | Parrot(string name, int year) :Animal(name, year) {} |
|  | Parrot() :Animal() {} |
|  | void golos() { |
|  | cout << "karrrr!!!" << endl; |
|  | } |
|  |  |
|  | }; |
|  | int main(){ |
|  | Cat a; |
|  | a.get\_Data(); |
|  | a.golos(); |
|  |  |
|  | Dog b; |
|  | b.get\_Data(); |
|  | b.golos(); |
|  |  |
|  | Parrot c; |
|  | c.get\_Data(); |
|  | c.golos(); |
|  |  |
|  |  |
|  |  |
|  |  |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Программа №2

|  |
| --- |
|  |
|  | #include <iostream> |
|  | #include <cmath> |
|  | using namespace std; |
|  |  |
|  |  |
|  | class Forug{ |
|  | protected: |
|  | int a1, a2, a3, a4; |
|  | public: |
|  | Forug() { |
|  | cout << "vvedite dliny storon" << endl; |
|  | cin >> a1 >> a2 >> a3 >> a4; |
|  | } |
|  | virtual void sqaire() = 0; |
|  |  |
|  |  |
|  | }; |
|  | class Romb:public Forug { |
|  | public: |
|  | double sinus; |
|  | Romb() :Forug() { |
|  | cout << "vvedite men'shiy ugol" << endl; |
|  | cin >> sinus; |
|  | } |
|  | void sqaire() { |
|  | cout << a1\*a1\*sin(sinus)<<endl;; |
|  | } |
|  | }; |
|  |  |
|  | class Pryamougolnic :public Forug { |
|  | public: |
|  | Pryamougolnic() :Forug() { |
|  |  |
|  | } |
|  | void sqaire() { |
|  | cout<< a1\*a3<<endl; |
|  | } |
|  | }; |
|  | class Figure { |
|  | public: |
|  |  |
|  | void sqaire(Forug \*forug) { |
|  | cout << "ploshad ravna:"; |
|  | forug->sqaire(); |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | int main() |
|  | { |
|  | Pryamougolnic c; |
|  | Romb b; |
|  | Figure a; |
|  | a.sqaire(&b); |
|  | a.sqaire(&c); |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Программа №3

|  |
| --- |
|  |
|  | #include <iostream> |
|  | #include <cmath> |
|  | #include <string> |
|  | using namespace std; |
|  |  |
|  | class Vehicle { |
|  | protected:int massa; |
|  | int skorost; |
|  | string name; |
|  | public:Vehicle() { |
|  | cout << "vvedite massu skorost i nazvanie" << endl; |
|  | cin >> massa >> skorost >> name; |
|  | } |
|  | virtual void kto\_ya() = 0; |
|  | }; |
|  | class Bycicle:public Vehicle { |
|  | public: |
|  | Bycicle() :Vehicle() { } |
|  | void kto\_ya() { |
|  | cout << "ya velosiped:" << name << ' ' << skorost << "km/h " << massa << "kg"<<endl; |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | class Bus :public Vehicle { |
|  | public: |
|  | Bus() :Vehicle() {} |
|  | void kto\_ya() { |
|  | cout << "ya avtobus:" << name << ' ' << skorost << "km/h " << massa << "kg" << endl; |
|  | } |
|  |  |
|  | }; |
|  |  |
|  | class Avto :public Vehicle { |
|  | public: |
|  | Avto() :Vehicle() { } |
|  | void kto\_ya() { |
|  | cout << "ya mashina:" << name << ' ' << skorost << "km/h " << massa << "kg" << endl; |
|  | } |
|  |  |
|  | }; |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | int main() |
|  | { |
|  | string z; |
|  | cout << "vvedite transportnoe sredstdo (bus, bycicle, avto)" << endl; |
|  | cin >> z; |
|  | if(z=="bus"){ Bus a; a.kto\_ya(); |
|  | } |
|  | else if (z == "avto") { |
|  | Avto a; a.kto\_ya(); |
|  | } |
|  | else if (z == "bycicle") { |
|  | Bycicle a; a.kto\_ya(); |
|  | } |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Программа №4

|  |
| --- |
|  |
|  | #include <iostream> |
|  | using namespace std; |
|  |  |
|  | class Screen { // Базовый абстрактный класс "Экран" |
|  | protected: |
|  | double razr; |
|  | public: |
|  | Screen(double size) { this->razr = size; } |
|  |  |
|  | virtual void izmScreen(double size) = 0; // Виртуальный метод заены экрана |
|  |  |
|  |  |
|  | }; |
|  |  |
|  | class Keyboard { // Базовый абстрактный класс "Клавиатура" |
|  | protected: |
|  | int buttons; |
|  | public: |
|  | Keyboard(int buttons) { this->buttons = buttons; } |
|  |  |
|  | virtual void izmKeyboard(int buttons) = 0; // Виртуальный метод замены клавиатуры |
|  | }; |
|  |  |
|  | class Computer : public Screen, Keyboard { // Класс "Стационарный компьютер" |
|  | public: |
|  | Computer(double razr, int buttons) : Screen(razr), Keyboard(buttons) {}; |
|  |  |
|  | void izmScreen(double size) { |
|  | cout << "ekran " << razr << |
|  | " byl izmenen na " << size << " razmer" << endl; |
|  | razr = size; |
|  | } |
|  |  |
|  | void izmKeyboard(int buttons) { |
|  | cout << "klaviatura s " << this->buttons << " knopkami " << |
|  | "byla izmenena na " << buttons << " knopok" << endl; |
|  | this->buttons = buttons; |
|  | } |
|  |  |
|  | ~Computer() {} |
|  | }; |
|  |  |
|  | class Notebook : public Screen, Keyboard { // Класс "Ноутбук" |
|  | public: |
|  | Notebook(double razr, int buttons) : Screen(razr), Keyboard(buttons) {}; |
|  |  |
|  | void izmScreen(double size) { |
|  | if (razr == size) { |
|  | cout << "ekran byl izmenen" << endl; |
|  | } |
|  | else { |
|  | cout << "ekran nouta ne byl izmenen" << endl; |
|  | } |
|  | } |
|  |  |
|  | void izmKeyboard(int buttons) { |
|  | if (this->buttons == buttons) { |
|  | cout << "klaviatura byla izmenena" << endl; |
|  | } |
|  | else { |
|  | cout << "klaviatura u niutbuka ne byla izmenena" << endl; |
|  | } |
|  | } |
|  |  |
|  | ~Notebook() {} |
|  | }; |
|  |  |
|  | class Phone : public Screen, Keyboard { // Класс "Телефон" |
|  | public: |
|  | Phone(double razr, int buttons) : Screen(razr), Keyboard(buttons) {}; |
|  |  |
|  | void izmScreen(double size) { |
|  | if (razr == size) { |
|  | cout << "ekran telefona byl izmenen" << endl; |
|  | } |
|  | else { |
|  | cout << "ekran telefona ne byl izmenen" << endl; |
|  | } |
|  | } |
|  |  |
|  | void izmKeyboard(int buttons) { |
|  | cout << "klaviatura telefona ne mozet byt izmenena" << endl; |
|  | } |
|  |  |
|  | ~Phone() {} |
|  | }; |
|  |  |
|  | int main() { |
|  |  |
|  | int z=1; |
|  | do { |
|  | cout << "s kakim ustroistvom budem rabokat 1-telefon, 2- notebook, 3- computer" << endl; |
|  | int e, r; |
|  | cin >> z; |
|  | switch (z) |
|  | { |
|  | case 1: {cout << "vvedite kolvo razmer ekrana i kolichestvo knopok" << endl; |
|  |  |
|  | cin >> e >> r; |
|  | Phone a(e, r); |
|  | cout << "vvedize na skol'ko vy hotite izmenit razmer ekrana i kolichestvo knopok" << endl; |
|  | cin >> e >> r; |
|  | a.izmKeyboard(e); |
|  | a.izmScreen(r); |
|  | break; } |
|  | case 2: {cout << "vvedite kolvo razmer ekrana i kolichestvo knopok" << endl; |
|  |  |
|  | cin >> e >> r; |
|  | Notebook b(e, r); |
|  | cout << "vvedize na skol'ko vy hotite izmenit razmer ekrana i kolichestvo knopok" << endl; |
|  | cin >> e >> r; |
|  | b.izmKeyboard(e); |
|  | b.izmScreen(r); |
|  | break; } |
|  | case 3: {cout << "vvedite kolvo razmer ekrana i kolichestvo knopok" << endl; |
|  |  |
|  | cin >> r >> e; |
|  | Computer c(e, r); |
|  | cout << "vvedize na skol'ko vy hotite izmenit razmer ekrana i kolichestvo knopok" << endl; |
|  | cin >> e >> r; |
|  | c.izmKeyboard(e); |
|  | c.izmScreen(r); |
|  | break; } |
|  | default: |
|  | break; |
|  | } |
|  |  |
|  | } while (z != 0); |
|  | system("pause"); |
|  | return 0; |
|  | } |

**Вывод**

Все задачи были выполнены. В ходе выполнения работы были более плотно изучены множественное наследование и абстрактные классы. Для достижения поставленной цели были использованы различные STL контейнеры и библиотека cmath. Результаты работы программ представлены на скриншотах (рис. 21-24). Исходный кодвыполненных програм также доступе но ссылка на GitHub:

1. [<https://github.com/dimaINBO0418/oop/blob/master/7.1.txt>](https://github.com/VladRubel/laby/blob/master/7.1)
2. [[https://github.com/dimaINBO0418/oop/blob/master/7.2.txt](https://github.com/VladRubel/laby/blob/master/7.1)](https://github.com/VladRubel/laby/blob/master/7.2)
3. [[https://github.com/dimaINBO0418/oop/blob/master/7.3.txt](https://github.com/VladRubel/laby/blob/master/7.1)](https://github.com/VladRubel/laby/blob/master/7.3)
4. [[https://github.com/dimaINBO0418/oop/blob/master/7.4.txt](https://github.com/VladRubel/laby/blob/master/7.4)](https://github.com/VladRubel/laby/blob/master/7.1)

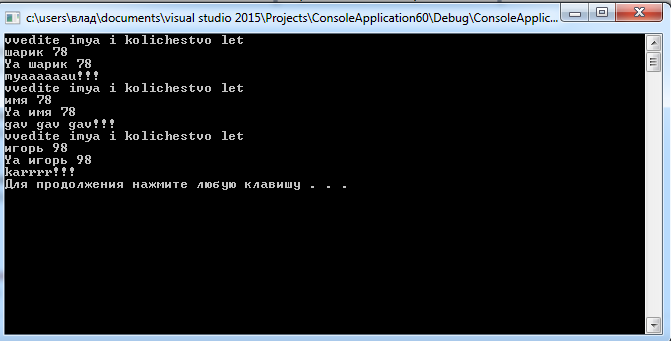
****

Рис. 21 Результат работы программы №1

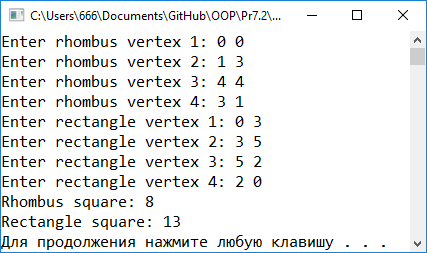


Рис. 22 Результат работы программы №2

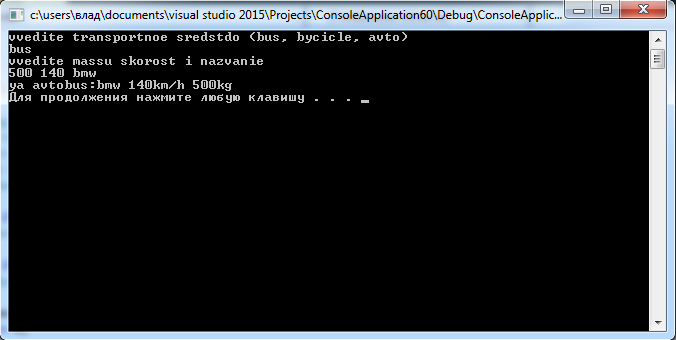


Рис. 23 Результат работы программы №3

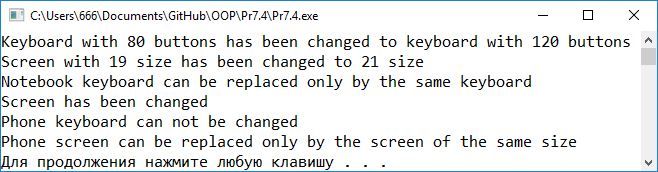


Рис. 24 Результат работы программы №4

## 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**

1. Программа №1

|  |
| --- |
|  |
|  | #include <fstream> |
|  | #include <iostream> |
|  | #include <string> |
|  | using namespace std; |
|  |  |
|  | int main(int argc, char\* argv[]) |
|  | { |
|  | string s; |
|  | cin >> s; |
|  | ofstream fout("fff.txt"); // создаём объект класса ofstream для записи и связываем его с файлом cppstudio.txt |
|  | fout << s; // запись строки в файл |
|  | fout.close(); // закрываем файл |
|  | system("pause"); |
|  | return 0; |
|  | } |

1. Программа №2

|  |
| --- |
|  |
|  | #include <fstream> |
|  | #include <iostream> |
|  | #include <string> |
|  | using namespace std; |
|  |  |
|  | int main() |
|  | { |
|  | setlocale(LC\_ALL, "rus"); |
|  | string s; |
|  | ifstream file("fff.txt"); |
|  |  |
|  | if (file.is\_open()){ // вызов метода is\_open() |
|  | cout << "Все ОК! Файл открыт!\n\n" << endl; |
|  | for (file >> s; !file.eof(); file >> s) |
|  | cout << s << endl; |
|  | } |
|  | else |
|  | { |
|  | cout << "Файл не открыт!\n\n" << endl; |
|  | return -1; |
|  | } |
|  | system("pause"); |
|  | return 0; |
|  | } |

**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. Результаты работ программ приведены на скриншотах (рис. 25, 26). Также исходный код програм доступенн по ссылкам на GitHub:

1. <https://github.com/dimaINBO0418/oop/blob/master/8.1.txt>
2. <https://github.com/dimaINBO0418/oop/blob/master/8.2.txt>

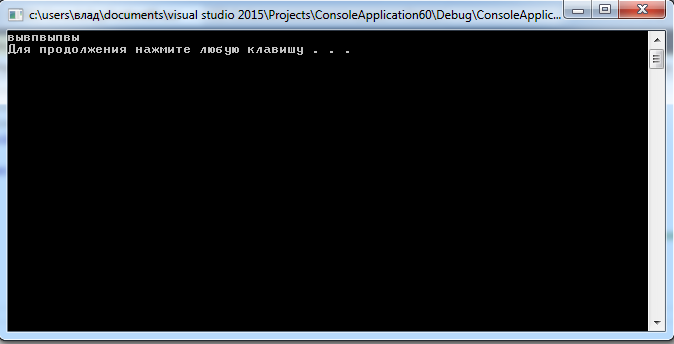


Рис. 25 Результат работы программы №1

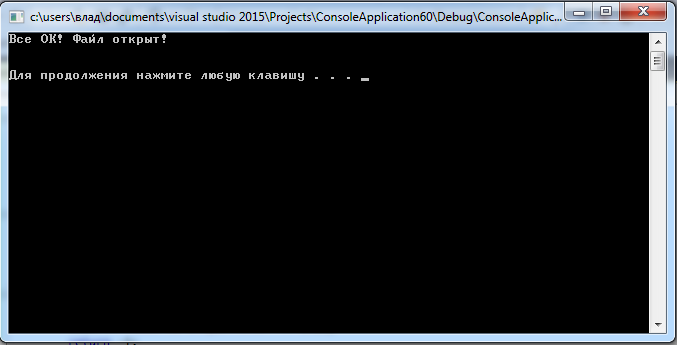


Рис. 26 Результат работы программы №2