

**Московский авиационный институт
(национальный исследовательский университет)**

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу «ООП»

Студент: Носов Э. Л.
Преподаватель:
Группа: М8О-308Б-21
Дата:
Оценка:
Подпись:

Москва, 2024

Лабораторная работа №3

Цель:

- Программирование классов на языке C++;
- Управление памятью в языке C++;
- Изучение базовых понятий ООП;
- Знакомство с классами в C++;
- Знакомство с перегрузкой операторов;
- Знакомство с дружественными функциями;
- Знакомство с операциями ввода-вывода из стандартных библиотек.

Вариант №29: Необходимо спроектировать и запрограммировать на языке C++ классы трех фигур: треугольник, квадрат, прямоугольник. Классы должны удовлетворять следующим правилам:

- Должны быть названы также, как в вариантах задания и расположены в отдельных файлах: отдельно заголовки (имя_класса_с_маленькой_буквы.h), отдельно описания методов (имя_класса_с_маленькой_буквы.cpp);
- Иметь общий родительский класс Figure;
- Содержать конструктор по умолчанию;
- Содержать конструктор, принимающий координаты вершин фигуры из стандартного потока `std::cin`, расположенных через пробел. Пример: 0.0 0.0 1.0 0.0 1.0 1.0 0.0 1.0
- Содержать набор общих методов:
 - `size_t VertexesNumber()` – метод, возвращающий количество вершин фигуры;
 - `double Area()` – метод расчета площади фигуры;
 - `void Print(std::ostream& os)` – метод печати типа фигуры и ее координат вершин в поток вывода `os` в формате:
Rectangle: (0.0, 0.0) (1.0, 0.0) (1.0, 1.0) (0.0, 1.0)\n

Программа должна позволять:

- Вводить произвольные фигуры и добавлять их в общий контейнер. Разрешается использовать стандартный контейнер `std::vector`;
- Распечатывать содержимое контейнера;

Ссылка на репозиторий: https://github.com/nsveml/UniProjects/tree/main/OOP/oop_exercise_3

1 Исходный код

В заголовочном файле figure.h определяется материнский класс Figure и объявляются его методы, от которого затем наследуются соответствующие классы в файлах triangle.h, square.h и rectangle.h

В файле figure.cpp происходит определение методов класса Figure. В файлах triangle.cpp, square.cpp, rectangle.cpp переопределяются конструкторы соответствующих классов.

В файле main.cpp реализовано взаимодействие с пользователем, позволяющее создавать фигуры заданных классов и помещать их в контейнер vector.

figure.h

```
1 | #include<iostream>
2 |
3 | #ifndef figure_h
4 | #define figure_h
5 |
6 | class Figure{
7 |     public:
8 |
9 |     Figure();
10 |    Figure(int a);
11 |    virtual ~Figure();
12 |    virtual size_t VertexesNumber();
13 |    virtual float Area();
14 |    virtual void Print(std::ostream& os);
15 |
16 |    protected:
17 |
18 |    size_t n;
19 |    float* coords;
20 |    const char* name = "Figure";
21 |
22 | };
23 |
24 | #include"figure.cpp"
25 |
26 | #endif
```

figure.cpp

```
1 | Figure::Figure(){
2 |     n=0;
3 |     coords=nullptr;
4 | }
5 |
6 | Figure::Figure(int a){
7 |     n=a;
8 |     coords = new float [2*n];
9 |     std::cout<<"Enter coordinates in clockwise order:\n";
10 |    for(long unsigned int i = 0; i<2*n; i++){
11 |        std::cin>>coords[i];
12 |    }
13 | }
14 |
15 | Figure::~Figure(){
16 |     delete [] coords;
17 | }
```

```

18 |
19 | size_t Figure::VertexesNumber(){
20 |     return n;
21 | }
22 |
23 | float Figure::Area(){
24 |     float area = 0;
25 |     for(long unsigned int i=0; i<2*n; i++){
26 |         area += (i % 2) ? -coords[i]*coords[(i+1) % (2*n)] : coords[i]*coords[(i+3) % (2*n)];
27 |     }
28 |     area/=2;
29 |     return (area<0)? -area : area;
30 | }
31 |
32 | void Figure::Print(std::ostream& os){
33 |     os<<name<<": ";
34 |     for(long unsigned int i = 0; i<2*n; i+=2){
35 |         os<<" ("<<coords[i]<<" , "<<coords[i+1]<<')';
36 |     }
37 |     os<<'\n';
38 | }

```

triangle.h

```

1 | #include"figure.h"
2 |
3 | class Triangle : public Figure{
4 |     public:
5 |
6 |     Triangle();
7 |
8 |     Triangle(std::istream& is);
9 |
10 | };
11 |
12 | #include"triangle.cpp"

```

triangle.cpp

```

1 | Triangle::Triangle(){
2 |     name = "Triangle";
3 |     n=3;
4 |     coords = new float [2*n];
5 |     for(long unsigned int i = 0; i<2*n; i++){
6 |         coords[i]=0.0;
7 |     }
8 | }
9 |
10 | Triangle::Triangle(std::istream& is){
11 |     name = "Triangle";
12 |     n=3;
13 |     coords = new float[2*n];
14 |     for(long unsigned int i=0; i<2*n; i++){
15 |         is>>coords[i];
16 |     }
17 | }

```

square.h

```

1 | #include"figure.h"
2 |
3 | class Square : public Figure{
4 |     public:

```

```

5 |
6 |     Square();
7 |
8 |     Square(std::istream& is);
9 |
10 | };
11 |
12 | #include "square.cpp"

```

square.cpp

```

1 | Square::Square(){
2 |     name = "Square";
3 |     n=4;
4 |     coords = new float [2*n];
5 |     for(long unsigned int i = 0; i<2*n; i++){
6 |         coords[i]=0.0;
7 |     }
8 | }
9 |
10 | Square::Square(std::istream& is){
11 |     name = "Square";
12 |     n=4;
13 |     coords = new float [2*n];
14 |     for(long unsigned int i=0; i<2*n; i++){
15 |         is>>coords[i];
16 |     }
17 | }

```

rectangle.h

```

1 | #include "figure.h"
2 |
3 | class Rectangle : public Figure{
4 |     public:
5 |
6 |     Rectangle();
7 |
8 |     Rectangle(std::istream& is);
9 |
10 | };
11 |
12 | #include "rectangle.cpp"

```

rectangle.cpp

```

1 | Rectangle::Rectangle(){
2 |     name = "Rectangle";
3 |     n=4;
4 |     coords = new float [2*n];
5 |     for(long unsigned int i = 0; i<2*n; i++){
6 |         coords[i]=0.0;
7 |     }
8 | }
9 |
10 | Rectangle::Rectangle(std::istream& is){
11 |     name = "Rectangle";
12 |     n=4;
13 |     coords = new float [2*n];
14 |     for(long unsigned int i=0; i<2*n; i++){
15 |         is>>coords[i];
16 |     }
17 | }

```

main.cpp

```
1  #include<iostream>
2  #include<vector>
3  #include"rectangle.h"
4  #include"triangle.h"
5  #include"square.h"
6
7  int main(){
8      std::cout<<"\n
9      t - create triangle\n\
10     s - create square\n\
11     r - create rectangle\n\
12     p - print information\n\
13     q - quit\n";
14
15     std::vector<Figure*> figures;
16     Figure* figure = nullptr;
17     char a;
18
19     while(true){
20         std::cout << "Enter commands:\n";
21         std::cin>>a;
22         switch(a){
23             case 't':
24                 std::cout << "Enter coordinates in clockwise order:\n";
25                 figure = new Triangle(std::cin);
26                 figures.push_back(figure);
27                 break;
28             case 's':
29                 std::cout << "Enter coordinates in clockwise order:\n";
30                 figure = new Square(std::cin);
31                 figures.push_back(figure);
32                 break;
33             case 'r':
34                 std::cout << "Enter coordinates in clockwise order:\n";
35                 figure = new Rectangle(std::cin);
36                 figures.push_back(figure);
37                 break;
38             case 'p':
39                 for(long unsigned int i = 0; i<figures.size(); i++){
40                     std::cout << "Number of vertexes:" << figures[i]->VertexesNumber() << "\n";
41                     std::cout << "Coordinates:";
42                     figures[i]->Print(std::cout);
43                     std::cout << "Area:" << figures[i]->Area() << "\n\n";
44                 }
45                 break;
46             case 'q':
47                 for(long unsigned int i = 0; i < figures.size(); i++){
48                     delete figures[i];
49                 }
50                 return 0;
51                 break;
52         }
53     }
54 }
55 }
```

2 Тестовые данные

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ cat test_01
```

t

1 0

0 0

0 1

s

0 0

1 0

1 1

0 1

r

0 0

1 0

1 2

0 2

p

```
qnsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ cat test_02
```

s

-1 0

0 1

1 0

0 -1

r

0 0

2 2

3 1

-1 -1

t

1 1

2 -1

0 0

p

3 Работа программы

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ mkdir cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ cd cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3/cm$ cmake ../
CMake Deprecation Warning at CMakeLists.txt:1 (cmake_minimum_required):
```

```
  Compatibility with CMake < 2.8.12 will be removed from a future version of
  CMake.
```

Update the VERSION argument <min> value or use a ...<max> suffix to tell
CMake that the project does not need compatibility with older versions.

```
-- The C compiler identification is GNU 12.0.1
-- The CXX compiler identification is GNU 12.0.1
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working C compiler: /usr/bin/cc - skipped
-- Detecting C compile features
-- Detecting C compile features - done
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++ - skipped
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Configuring done
-- Generating done
-- Build files have been written to: /mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3/cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3/cm$ make
[ 50%] Building CXX object CMakeFiles/oop_exercise_3.dir/main.cpp.o
[100%] Linking CXX executable ../oop_exercise_3
[100%] Built target oop_exercise_3
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3/cm$ cd ..
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ ls
CMakeLists.txt  cm  figure.cpp  figure.h  main.cpp  oop_exercise_3  rectangle.cpp  rectangle.h  square.c
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3$ ./oop_exercise_3 < test_01
    t - create triangle
    s - create square
    r - create rectangle
    p - print information
    q - quit
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Number of vertexes:3
```


Coordinates:Triangle: (1, 0) (0, 0) (0, 1)
Area:0.5

Number of vertexes:4
Coordinates:Square: (0, 0) (1, 0) (1, 1) (0, 1)
Area:1

Number of vertexes:4
Coordinates:Rectangle: (0, 0) (1, 0) (1, 2) (0, 2)
Area:2

Enter commands:
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/UniProjects/OOP/oop_exercise_3\$./oop_exercise_3 < test_02
t - create triangle
s - create square
r - create rectangle
p - print information
q - quit
Enter commands:
Enter coordinatesin clockwise order:
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Number of vertexes:4
Coordinates:Square: (-1, 0) (0, 1) (1, 0) (0, -1)
Area:2

Number of vertexes:4
Coordinates:Rectangle: (0, 0) (2, 2) (3, 1) (-1, -1)
Area:3

Number of vertexes:3
Coordinates:Triangle: (1, 1) (2, -1) (0, 0)
Area:1.5

Enter commands:

4 Вывод

Выполнив третью лабораторную работу по курсу «ООП», я изучил принцип наследования классов в языке C++ и закрепил полученные ранее знания и навыки.