

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

**Факультет информационных технологий и прикладной математики**

**Кафедра вычислительной математики и программирования**

**Лабораторная работа №7 по курсу «ООП»**

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

**Москва, 2024**

## Лабораторная работа №7

### Цель:

- Закрепление навыков работы с шаблонами классов;
- Построение итераторов для динамических структур данных

**Вариант №29:** Используя структуру данных, разработанную для лабораторной работы №4, спроектировать и разработать итератор для динамической структуры данных. Итератор должен быть разработан в виде шаблона и должен позволять работать с любыми типами фигур, согласно варианту задания. Итератор должен позволять использовать структуру данных в операторах типа for.

Программа должна позволять:

- Вводить произвольное количество фигур и добавлять их в контейнер;
- Распечатывать содержимое контейнера;
- Удалять фигуры из контейнера

Ссылка на репозиторий: [https://github.com/nsveml/UniProjects/tree/main/OOP/oop\\_exercise\\_7](https://github.com/nsveml/UniProjects/tree/main/OOP/oop_exercise_7)

7

# 1 Исходный код

Классы фигур реализованы так же как в предыдущих работах.

В заголовочном файле queue.h определены шаблоны классов контейнера TQueue и итератора Iterator. В классе TQueue указаны дружественный шаблон класса Iterator, приватные атрибуты: целочисленные size и capacity, шаблон умного указателя типа shared\_ptr storage, в нем же объявлены шаблоны методов класса, а именно публичные Top, Pop, Push, Empty, Length, Clear, begin, end, конструкторы и деструктор, приватный метод Extend, необходимый для реаллокации памяти, и перегрузка оператора вывода.

В классе Iterator указаны приватные атрибуты: size и index типа size\_t, указатель, тип которого задается параметром шаблона T. Методы этого класса объявлены в объеме, необходимом для использования цикла for для итерирования по контейнеру: конструктор и перегрузка операторов \*, ->, ++ и !=.

Определения всех функций, методов и перегрузок операторов шаблонов классов вынесены в файл queue.cpp.

В файле main.cpp реализовано взаимодействие с пользователем, позволяющее создавать фигуры заданного класса и помещать их в контейнер. Для вывода содержимого контейнера используется цикл for.

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 }

```

queue.h

```

1  #ifndef QUEUE_H
2  #define QUEUE_H
3
4  #include<memory>
5
6  template<class T>
7  class TQueue;
8
9  template<class T>
10 std::ostream& operator<< (std::ostream&, const TQueue<T>&);
11
12 template<class T>
13 class Iterator;
14
15 template <class T>
16 class TQueue {
17     friend class Iterator<T>;
18
19     public:
20     TQueue();
21     TQueue(const TQueue& other);
22     void Push(const T& polygon);
23     const T Pop();
24     const T& Top();
25     bool Empty();
26     size_t Length();
27     friend std::ostream& operator<< <> (std::ostream&, const TQueue&);
28     void Clear();
29     virtual ~TQueue();
30     Iterator<T> begin();
31     Iterator<T> end();
32
33     protected:
34     int size, capacity;
35     std::shared_ptr<T[]> storage;
36     void extend();
37 };
38
39 template <class T>
40 class Iterator{
41     T* array;
42     size_t size;
43     size_t index;
44     public:
45     Iterator(T* ptr, size_t i, size_t s);
46     T operator * ();
47     T operator -> ();
48     bool operator != (const Iterator<T> & other) const;
49     Iterator<T> & operator ++ ();
50 };
51
52 #include"queue.cpp"
53
54 #endif

```

queue.cpp

```
1 template<class T>
2 TQueue<T>::TQueue(){
3     size = 0;
4     capacity = 1;
5     storage.reset(new T[1]);
6 }
7
8 template<class T>
9 TQueue<T>::TQueue(const TQueue<T>& other){
10     size = other.size;
11     capacity = other.capacity;
12     storage = std::make_shared<T[]>(capacity);
13     for(int i = 0; i < size; i++){
14         storage[i]=other.storage[i];
15     }
16 }
17
18 template<class T>
19 void TQueue<T>::Push(const T& polygon){
20     if(capacity == size) extend();
21     storage[size]=polygon;
22     size++;
23 }
24
25 template<class T>
26 const T TQueue<T>::Pop(){
27     if(size > 0){
28         size--;
29         return(storage[size]);
30     }
31     else{
32         return T();
33     }
34 }
35
36 template<class T>
37 const T& TQueue<T>::Top(){
38     return storage[size-1];
39 }
40
41 template<class T>
42 bool TQueue<T>::Empty(){
43     return bool(size);
44 }
45
46 template<class T>
47 size_t TQueue<T>::Length(){
48     return size;
49 }
50
51 template<class T>
52 std::ostream& operator<< (std::ostream& os, const TQueue<T>& queue){
53     os<<"Queue:\nsize: "<<queue.size<<"\ncapacity:"<< queue.capacity<<"\n"<<"=> ";
54     for(int i = 0; i < queue.size; i++){
55         os << queue.storage[i].Area() << ' ';
56     }
57     os << "=>\n";
58     return os;
59 }
60
61 template<class T>
```

```

62 void TQueue<T>::Clear(){
63     size = 0;
64     capacity = 1;
65     storage.reset(new T[1]);
66 }
67
68 template<class T>
69 TQueue<T>::~TQueue(){
70     size = 0;
71     capacity = 0;
72     storage.reset();
73 }
74
75 template<class T>
76 void TQueue<T>::extend(){
77     T* tmp;
78     capacity*=2;
79     tmp = new T[capacity];
80     for(int i = 0; i < size; i++){
81         tmp[i]=storage[i];
82     }
83     storage.reset();
84     storage.reset(tmp);
85 }
86
87 template<class T>
88 Iterator<T> TQueue<T>::begin(){
89     return Iterator(storage.get(), 0, size);
90 }
91
92 template<class T>
93 Iterator<T> TQueue<T>::end(){
94     return Iterator(storage.get()+size, size, size);
95 }
96
97 template<class T>
98 Iterator<T>::Iterator(T* ptr, size_t i,size_t s){
99     array =ptr;
100     index= i;
101     size = s;
102 }
103
104 template<class T>
105 T Iterator<T>::operator * (){
106     return *array;
107 }
108
109 template<class T>
110 T Iterator<T>::operator -> (){
111     return *array;
112 }
113
114 template<class T>
115 bool Iterator<T>::operator != (const Iterator<T> & other) const{
116     return array != other.array;
117 }
118
119 template<class T>
120 Iterator<T> & Iterator<T>::operator ++ (){
121     this->array++;
122     return *this;
123 }

```

main.cpp

```
1  #include<iostream>
2  #include"triangle.h"
3  #include"rectangle.h"
4  #include"square.h"
5  #include"queue.h"
6
7  int main(){
8      std::cout<<"\n
9      u - push\n\
10     o - pop\n\
11     t - top\n\
12     c - clear\n\
13     i - print\n\
14     q - quit\n";
15
16     Triangle triangle;
17     TQueue<Triangle> triangles;
18     char a=' ';
19
20     while(a!='q'){
21         std::cout << "Enter commands:\n";
22         std::cin>>a;
23         switch(a){
24             case 'u':
25                 std::cout << "Enter coordinates in clockwise order:\n";
26                 std::cin >> triangle;
27                 triangles.Push(triangle);
28                 break;
29             case 'o':
30                 std::cout << triangles.Pop();
31                 break;
32             case 't':
33                 std::cout << triangles.Top();
34                 break;
35             case 'c':
36                 triangles.Clear();
37                 std::cout << "Cleared\n";
38                 break;
39             case 'i':
40                 for(auto i : triangles) {
41                     std::cout << i << std::endl;
42                 }
43                 break;
44             case 'q':
45                 break;
46         }
47     }
48 }
```



## 2 Тестовые данные

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ cat test1
```

```
u
1 1
0 0
1 0
u
2 2
0 0
1 0
u
3 3
0 0
1 0
u
4 4
0 0
1 0
i
q
```

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ cat test2
```

```
u
1 1
0 0
1 0
t
u
2 2
0 0
1 0
t
u
3 3
0 0
1 0
t
i
u
4 4
0 0
1 0
t
i
o
i
q
```

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ cat test3
```

```
u
```

1 1  
0 0  
1 0  
t  
u  
2 2  
0 0  
1 0  
t  
u  
3 3  
0 0  
1 0  
t  
u  
4 4  
0 0  
1 0  
t  
i  
c  
i  
q

### 3 Работа программы

```
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ mkdir cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ cd cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7/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/3 sem/OOP/labs/oop_exercise_7/cm
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7/cm$ make
[ 50%] Building CXX object CMakeFiles/oop_exercise_7.dir/main.cpp.o
[100%] Linking CXX executable ../oop_exercise_7
[100%] Built target oop_exercise_7
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7/cm$ cd ..
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ ls
CMakeLists.txt  cm  headers  main.cpp  oop_exercise_7  test1  test2  test3
nsveml@LAPTOP-LNCHGOM3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ ./oop_exercise_7 < test1
  u - push
  o - pop
  t - top
  c - clear
  i - print
  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:
```

Enter coordinates in clockwise order:

Enter commands:

Figure: (1, 1) (0, 0) (1, 0)

Figure: (2, 2) (0, 0) (1, 0)

Figure: (3, 3) (0, 0) (1, 0)

Figure: (4, 4) (0, 0) (1, 0)

Enter commands:

nsveuml@LAPTOP-LNCHG0M3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop\_exercise\_7\$ ./oop\_exercise\_7 < test2

u - push

o - pop

t - top

c - clear

i - print

q - quit

Enter commands:

Enter coordinates in clockwise order:

Enter commands:

Triangle: (1, 1) (0, 0) (1, 0)

Enter commands:

Enter coordinates in clockwise order:

Enter commands:

Triangle: (2, 2) (0, 0) (1, 0)

Enter commands:

Enter coordinates in clockwise order:

Enter commands:

Triangle: (3, 3) (0, 0) (1, 0)

Enter commands:

Figure: (1, 1) (0, 0) (1, 0)

Figure: (2, 2) (0, 0) (1, 0)

Figure: (3, 3) (0, 0) (1, 0)

Enter commands:

Enter coordinates in clockwise order:

Enter commands:

Triangle: (4, 4) (0, 0) (1, 0)

Enter commands:

Figure: (1, 1) (0, 0) (1, 0)

Figure: (2, 2) (0, 0) (1, 0)

Figure: (3, 3) (0, 0) (1, 0)

Figure: (4, 4) (0, 0) (1, 0)

```

Enter commands:
Figure: (4, 4) (0, 0) (1, 0)
Enter commands:
Figure: (1, 1) (0, 0) (1, 0)

Figure: (2, 2) (0, 0) (1, 0)

Figure: (3, 3) (0, 0) (1, 0)

Enter commands:
nsveml@LAPTOP-LNCHG0M3:/mnt/d/X-Files/MAI/3 sem/OOP/labs/oop_exercise_7$ ./oop_exercise_7 < test3
  u - push
  o - pop
  t - top
  c - clear
  i - print
  q - quit
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Triangle: (1, 1) (0, 0) (1, 0)
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Triangle: (2, 2) (0, 0) (1, 0)
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Triangle: (3, 3) (0, 0) (1, 0)
Enter commands:
Enter coordinates in clockwise order:
Enter commands:
Triangle: (4, 4) (0, 0) (1, 0)
Enter commands:
Figure: (1, 1) (0, 0) (1, 0)

Figure: (2, 2) (0, 0) (1, 0)

Figure: (3, 3) (0, 0) (1, 0)

Figure: (4, 4) (0, 0) (1, 0)

Enter commands:
Cleared
Enter commands:
Enter commands:

```

## 4 Вывод

Выполнив седьмую лабораторную работу по курсу «ООП», я закрепил полученные ранее знания, научился создавать и использовать итераторы для шаблонов классов, а так же реализовал собственный итератор для класса из предыдущей лабораторной работы.