



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 8
по курсу «Языки и методы программирования»
«Разработка шаблона класса»

Студент группы ИУ9-21Б Горбунов А. Д.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Согласно выбранному из таблиц 1–16 описанию требуется составить шаблон класса, разместив его в отдельном заголовочном файле. Проверку работоспособности класса требуется организовать в функции `main`, размещённой в файле «`main.cpp`».

`PascalArray<int L,int R,class T>` – массив с элементами типа `T`, индексируемые от `L` до `R`, с перегруженной операцией индексации и операцией конкатенации двух массивов, которая допустима в случае, если правая граница первого массива на единицу меньше левой границы второго массива. Массив, в котором $R = L - 1$, считается пустым и не обладает операцией индексации.

2 Результаты

Исходный код программы представлен в листинге 1, 2, 3

Листинг 1 — Функции для проверки(`main.cpp`)

```
1 #include<iostream>
2 #include<cassert>
3 #include "PascalArray.h"
4 using namespace std;
5 void test_index_operator() {
6     PascalArray<1, 5, int> p_array;
7     p_array[1] = 10;
8     p_array[2] = 20;
9     assert(p_array[1] == 10);
10    assert(p_array[2] == 20);
11 }
12 void test_concatenation_operator() {
13     PascalArray<1, 5, int> p_array1;
14     p_array1[1] = 10;
15     p_array1[2] = 20;
16     PascalArray<6, 9, int> p_array2;
17     p_array2[6] = 30;
18     p_array2[7] = 40;
19     p_array2[8] = 50;
20     p_array2[9] = 60;
21     auto p_array3 = p_array1 + p_array2;
22     assert(p_array3[1] == 10);
23     assert(p_array3[2] == 20);
24     assert(p_array3[6] == 30);
25     assert(p_array3[7] == 40);
26     assert(p_array3[8] == 50);
27     assert(p_array3[9] == 60);
28 }
```

Листинг 2 — main(main.cpp)

```
1 int main() {
2     test_index_operator();
3     test_concatenation_operator();
4     return 0;
5 }
```

Листинг 3 — класс PascalArray

```
1 template <int L, int R, class T>
2 class PascalArray {
3 private:
4     T arr[R - L + 1];
5 public:
6     T& operator[](int i) {
7         return arr[i - L];
8     }
9     template <int L2, int R2>
10    auto operator+ (PascalArray<L2, R2, T>& other) const {
11        static_assert(R == L2 - 1);
12        PascalArray<L, R2, T> result;
13        for (int i = L; i <= R2; ++i) {
14            if (i <= R) {
15                result[i] = arr[i - L];
16            } else {
17                result[i] = other[i];
18            }
19        }
20        return result;
21    }
22 };
```

Результат запуска представлен на рисунке 1, 2, 3

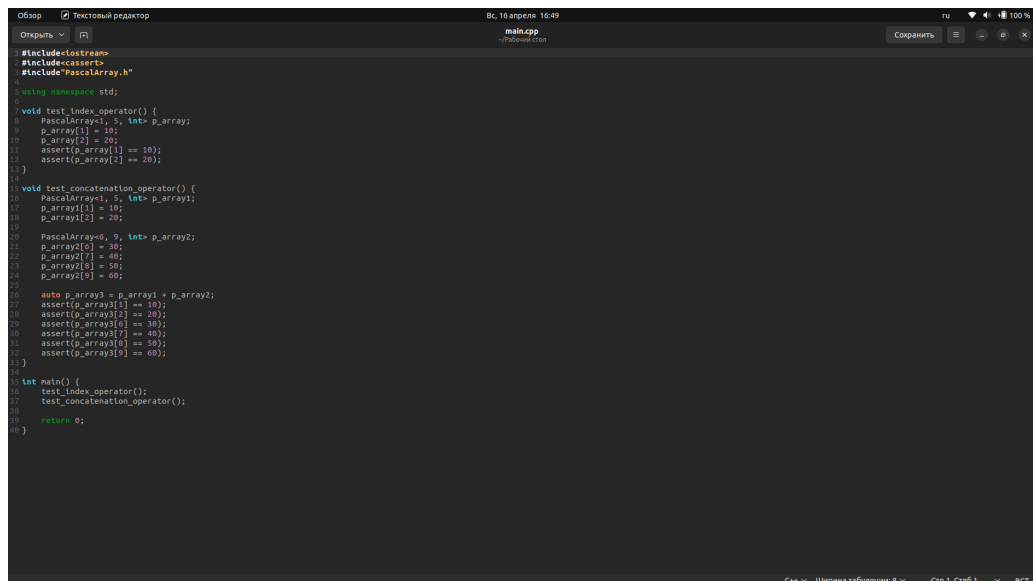


Рис. 1 — Реализация main.cpp

```
1 template<int L, int R, class T>
2 class PascalArray {
3 private:
4     T arr[R - L + 1];
5
6 public:
7     T& operator[](int l) {
8         return arr[l - L];
9     }
10
11     template<int L2, int R2>
12     auto operator+ (PascalArray<L2, R2, T>& other) const {
13         static const int R = L2 - L + 1;
14         PascalArray<L, R2, T> result;
15         for (int l = L; l <= R2; ++l) {
16             if (l <= R) {
17                 result[l] = arr[l - L];
18             } else {
19                 result[l] = other[l];
20             }
21         }
22         return result;
23     }
24 };
```

Рис. 2 — Реализация и PascalArray.h

```
1 #include<iostream>
2 #include<cassert>
3 #include<"PascalArray.h">
4 using namespace std;
5
6 void test_index_operator() {
7     PascalArray<1, 3, int> p_array;
8     p_array[1] = 10;
9     p_array[2] = 20;
10    assert(p_array[1] == 10);
11    assert(p_array[2] == 20);
12 }
13
14 void test_concatenation_operator() {
15     PascalArray<5, 10, int> p_array1;
16     p_array1[1] = 10;
17     p_array1[2] = 20;
18
19     PascalArray<9, 10, int> p_array2;
20     p_array2[3] = 30;
21     p_array2[7] = 40;
22     p_array2[8] = 50;
23     p_array2[9] = 60;
24
25     auto p_array3 = p_array1 + p_array2;
26     assert(p_array3[1] == 10);
27     assert(p_array3[2] == 20);
28     assert(p_array3[3] == 30);
29     assert(p_array3[7] == 40);
30     assert(p_array3[8] == 50);
31     assert(p_array3[9] == 60);
32 }
33
34 int main() {
35     test_index_operator();
36     test_concatenation_operator();
37     return 0;
38 }
```

Рис. 3 — Работа программы