Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №4 по теме «ШАБЛОНЫ ФУНКЦИЙ И КЛАССОВ»

По дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования»

Выполнил: Ролевич М.А.

Группа 19-ИТ-1.

Проверил: Васильева Д. М.

Новополоцк

2021 г.

**Вариант 3**

**Цель.** Получить практические навыки создания шаблонов и использования их в программах С++ / C# / Java.

**Основное содержание работы.**

Создать шаблон заданного класса и использовать его для данных различных типов.

**Порядок выполнения работы.**

1. Создать шаблон заданного класса. Определить конструкторы, деструктор, перегруженную операцию присваивания (“=”) и операции, заданные в варианте задания.

2. Написать программу тестирования, в которой проверяется использование шаблона для стандартных типов данных.

3. Выполнить тестирование.

4. Определить пользовательский класс, который будет использоваться в качестве параметра шаблона. Определить в классе необходимые функции и перегруженные операции.

5. Написать программу тестирования, в которой проверяется использование шаблона для пользовательского типа.

6. Выполнить тестирование.

Для моего варианта предложены следующие операции:

Класс − одномерный массив. Дополнительно перегрузить следующие операции:

[] − доступ по индексу;

== − проверка на равенство;

!= − проверка на неравенство.

**Задание 1. Определяем и реализовываем в классе конструкторы, деструктор, перегрузка операции:**

1. #include <string.h>
2. #include <iostream>
3. #include <tchar.h>
4. using namespace std;
5. template <class T>
6. class Array
7. {
8. private:
9. T\* mass;
10. int size;
11. int index;
12. int find(char sym) {
13. for (int i = 0; i < size; i++) {
14. if (this->setChar[i] == sym) {
15. return i;
16. }
17. }
18. return -1;
19. }
20. public:
21. Array(int n = 0); //конструктор по умолчанию
22. ~Array() {
23. if (mass) {
24. delete[] mass;
25. }
26. } //деструктор
27. Array(const Array& tam); //конструктор копирования
28. T& operator = (Array& tam) //перегрузка операции присваивания
29. T& operator != (Array& tam);//перегрузка операции на неравенство
30. const T& operator [] (int);//перегрузка оператора индексирования
31. void add();
32. void show\_list(); //вывод на экран
33. };

**Задание 2. Дополнить определение класса заданными перегруженными операциями:**

1. // конструктор
2. template <class T>Array<T>::Array(int n) {
3. size = n;
4. index = 0;
5. if (n)
6. {
7. mass = new T[size];
8. }
9. else
10. {
11. mass = 0;
12. }
13. }
14. //конструктор копирования
15. template<class T>Array<T>::Array(const Array& tam) {
16. size = tam.size;
17. index = 0;
18. if (tam.mass)
19. {
20. mass = new T[size];
21. for (int i = 0; i < size; i++)
22. {
23. mass[i] = tam.mass[i];
24. }
25. }
26. else
27. {
28. mass = 0;
29. }
30. }
31. template<class T>void Array<T>::show\_list() {
32. // вывод
33. for (int i = 0; i < index; i++) {
34. cout << mass[i] << " ";
35. }
36. cout << endl;
37. }
38. template<class T>const T& Array<T>::operator [] (int i) {
39. // доступ по индексу
40. if (i < size) {
41. return mass[i];
42. }
43. else {
44. cout << "Error" << endl;
45. exit(1);
46. }
47. }
48. template<class T>T& Array<T>::operator = (Array& tam) {
49. // присваивание
50. if (&tam == this) return \*this;//сравнение адресов
51. if (mass) delete[] mass;
52. size = tam.size;
53. if (tam.mass) {
54. mass = new T[size];
55. for (int i = 0; i < size; i++) {
56. mass[i] = tam.mass[i];
57. }
58. }
59. else
60. mass = 0;
61. return \*this;
62. }
63. template<class T>T& Array<T>::operator != (Array& tam) {
64. //на не равенства
65. for (int i = 0; i < this->size; i++) {
66. if (tam.find(this->mass[i]) == -1) {
67. return false;
68. }
69. }
70. return true;
71. }
72. template<class T>void Array<T>::add() {//добавление
73. T tamp;
74. for (int i = 0; i < size; i++) {
75. cout << "Введите значение: ";
76. cin >> tamp;
77. mass[i] = tamp;
78. ++index;
79. }
80. }

**Задание 3. Выполнить тестирование класса с перегрузками:**

1. int \_tmain(int argc, \_TCHAR\* argv[])
2. {
3. setlocale(LC\_ALL, "Russian");
4. Array<int>tam1(3);
5. cout << "char pressed" << endl;
6. tam1.add();
7. Array<int>tam2(3);
8. cout << "int pressed" << endl;
9. tam2.add();
10. Array<float>tam3(3);
11. cout << "float pressed" << endl;
12. tam3.add();
13. system("pause");
14. cout << "CHAR :" << endl;
15. tam1.show\_list();
16. cout << "INT :" << endl;
17. tam2.show\_list();
18. cout << "FLOAT :" << endl;
19. tam3.show\_list();
20. if (tam1 != tam2) {
21. cout << "Ravenstva ne ravni" << endl;
22. }
23. else {
24. cout << "Ravenstva ravni";
25. }
26. system("pause");
27. return 0;
28. }