Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №18.1**

Дисциплина: основы алгоритмизации и программирования

Тема: “Классы и объекты. Инкапсуляция”

Вариант 4

Выполнил:

Студент группы ИВТ-20-2б

Брейкин Алексей Дмитриевич

Проверил:

Доцент кафедры ИТАС

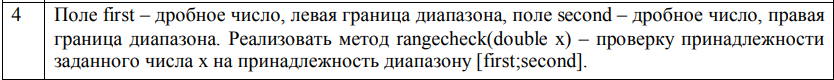
Полякова О. А.

**Пермь, 2020**

**Постановка задачи**

1. Реализовать определение нового класса. Продемонстрировать разные способы создания объектов и массивов объектов.

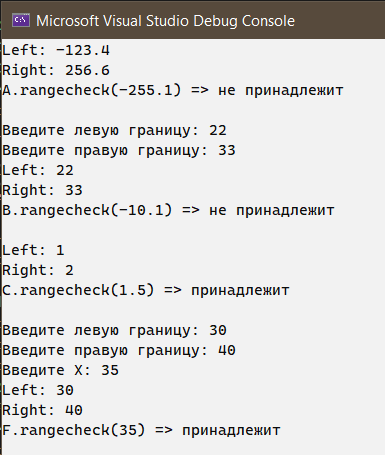
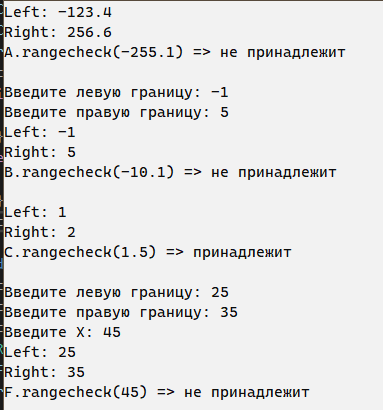
2. Структура-пара – структура с двумя полями, которые обычно имеют имена first и second.



**Исходный код**

1. #include <iostream>
3. using namespace std;
5. class Range {
6. double m\_left;
7. double m\_right;
8. public:
9. void Init(double left, double right) {
10. if (left > right) {
11. int tmp = left;
12. left = right;
13. right = tmp;
14. }
15. m\_left = left;
16. m\_right = right;
17. }
18. void Read() {
19. cout << "Введите левую границу: "; cin >> m\_left;
20. cout << "Введите правую границу: "; cin >> m\_right;
21. if (m\_left > m\_right) {
22. int tmp = m\_left;
23. m\_left = m\_right;
24. m\_right = tmp;
25. }
26. }
28. void Show() {
29. cout << "Left: " << m\_left << endl;
30. cout << "Right: " << m\_right << endl;
31. }
33. bool rangecheck(double x) {
34. return x >= m\_left && x <= m\_right;
35. }
36. };
38. Range make\_Range(double left, double right) {
39. Range result;
40. result.Init(left, right);
41. return result;
42. }
44. int main() {
45. system("chcp 1251>nul");
46. Range A;
47. A.Init(-123.4, 256.6);
48. A.Show();
49. bool res = A.rangecheck(-255.1);
50. cout << "A.rangecheck(-255.1) =>";
51. if (res) {
52. cout << " принадлежит" << endl;
53. }
54. else {
55. cout << " не принадлежит" << endl;
56. }
57. cout << endl;
59. Range B;
60. B.Read();
61. B.Show();
62. res = B.rangecheck(-10.1);
63. cout << "B.rangecheck(-10.1) =>";
64. if (res) {
65. cout << " принадлежит" << endl;
66. }
67. else {
68. cout << " не принадлежит" << endl;
69. }
70. cout << endl;
72. Range\* C = new Range;
73. C->Init(1, 2);
74. C->Show();
75. res = C->rangecheck(1.5);
76. cout << "C.rangecheck(1.5) =>";
77. if (res) {
78. cout << " принадлежит" << endl;
79. }
80. else {
81. cout << " не принадлежит" << endl;
82. }
83. cout << endl;
85. double left, right, x;
86. cout << "Введите левую границу: "; cin >> left;
87. cout << "Введите правую границу: "; cin >> right;
88. cout << "Введите X: "; cin >> x;
89. Range F = make\_Range(left, right);
90. F.Show();
91. res = F.rangecheck(x);
92. cout << "F.rangecheck(" << x << ") =>";
93. if (res) {
94. cout << " принадлежит" << endl;
95. }
96. else {
97. cout << " не принадлежит" << endl;
98. }
99. return 0;
100. }

**Скриншоты работы программы**

**** ****

**Контрольные вопросы**

1. Что такое класс?



1. Что такое объект (экземпляр) класса?

Переменная класса является объектом (экземпляром) класса. Класс объявляется один раз, а переменных может быть сколько угодно

1. Как называются поля класса?

Поля/атрибуты класса называются данными класса.

1. Как называются функции класса?

Функции класса называются методами.

1. Для чего используются спецификаторы доступа?

Спецификаторы доступа управляют видимостью элементов класса.

1. Для чего используется спецификатор public?

Спецификатор public используется, чтобы задать общедоступную видимость.

1. Для чего используется спецификатор private?

Спецификатор private используется, чтобы задать видимость только внутри класса для полей или методов под этим спецификатором.

1. Если описание класса начинается со спецификатора class, то какой спецификатор доступа будет использоваться по умолчанию?

Для спецификатора class по умолчанию используется private.

1. Если описание класса начинается со спецификатора struct, то какой спецификатор доступа будет использоваться по умолчанию?

Для спецификатора struct по умолчанию используется public.

1. Какой спецификатор доступа должен использоваться при описании интерфейса класса? Почему?

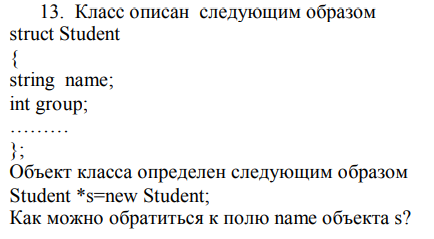
Интерфейс класса описывается после спецификатора public, чтобы к методам класса можно было обратится извне.

1. Каким образом можно изменять значения атрибутов экземпляров класса?

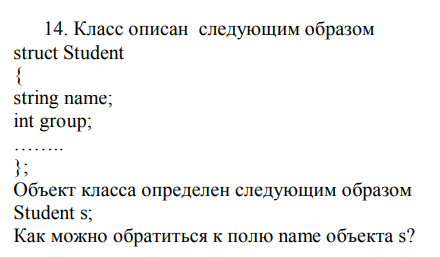
Значения атрибутов экземпляра класса можно изменить с помощью методов класса: модификаторов.

1. Каким образом можно получить значения атрибутов экземпляров класса?

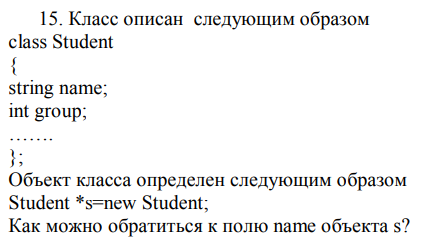
Значения атрибутов экземпляра класса можно получить с помощью методов класса: селекторов.



К полю name можно обратиться через оператор “->”.



К полю name можно обратиться через оператор “.”.



Чтобы обратиться к полю name, необходимо использовать спецификатор public.