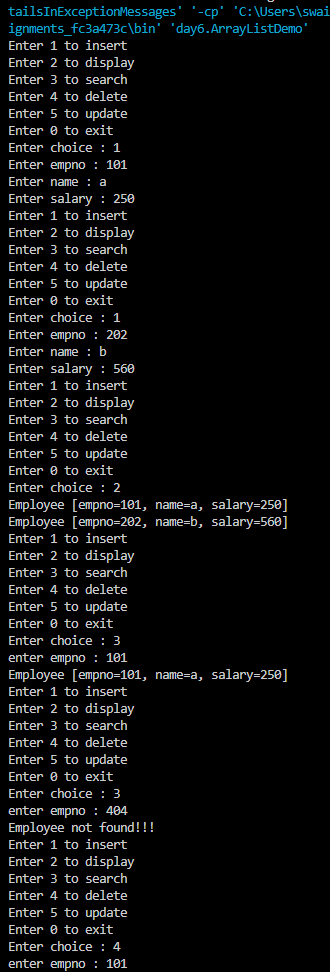
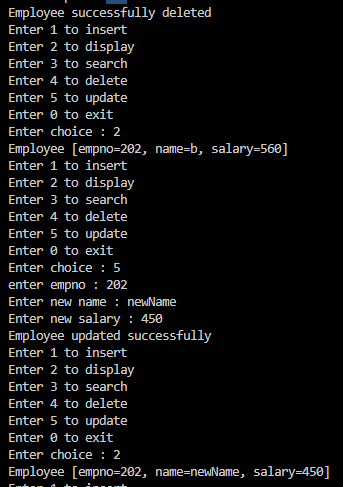
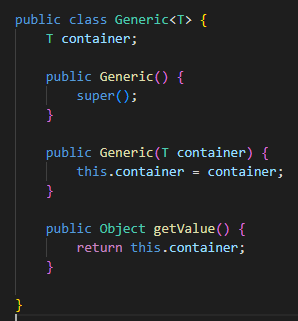
1. ArrayList Demo
2. import java.util.\*;
3. public class ArrayListDemo {
4. public static int find(List<Employee> e, int empno) {
5. int pos = -1;
6. for (int i = 0; i < e.size(); i++) {
7. Employee emp = e.get(i);
8. if (emp.getEmpno() == empno) {
9. pos = i;
10. break;
11. }
12. }
13. return pos;
14. }
15. public static void insert(List<Employee> e) {
16. Scanner sin = new Scanner(System.in);
17. int empno, salary;
18. String name;
19. System.out.print("Enter empno : ");
20. empno = Integer.parseInt(sin.nextLine());
21. System.out.print("Enter name : ");
22. name = sin.nextLine();
23. System.out.print("Enter salary : ");
24. salary = Integer.parseInt(sin.nextLine());
25. Employee emp = new Employee(empno, name, salary);
26. e.add(new Employee(empno, name, salary));
27. }
28. public static void display(List<Employee> e) {
29. Iterator itr = e.iterator();
30. while (itr.hasNext()) {
31. System.out.println(itr.next());
32. }
33. }
34. public static void search(List<Employee> e) {
35. Scanner sin = new Scanner(System.in);
36. int empno;
37. System.out.print("enter empno : ");
38. empno = Integer.parseInt(sin.nextLine());
39. ;
40. int pos = find(e, empno);
41. if (pos == -1) {
42. System.out.println("Employee not found!!!");
43. } else {
44. System.out.println(e.get(pos));
45. }
46. }
47. public static void delete(List<Employee> e) {
48. Scanner sin = new Scanner(System.in);
49. int empno;
50. System.out.print("enter empno : ");
51. empno = Integer.parseInt(sin.nextLine());
52. ;
53. int pos = find(e, empno);
54. if (pos == -1) {
55. System.out.println("Employee not found!!!");
56. } else {
57. e.remove(pos);
58. System.out.println("Employee successfully deleted");
59. }
60. }
61. public static void update(List<Employee> e) {
62. Scanner sin = new Scanner(System.in);
63. int empno;
64. System.out.print("enter empno : ");
65. empno = Integer.parseInt(sin.nextLine());
66. ;
67. int pos = find(e, empno);
68. if (pos == -1) {
69. System.out.println("Employee not found!!!");
70. } else {
71. String name;
72. int salary;
73. System.out.print("Enter new name : ");
74. name = sin.nextLine();
75. System.out.print("Enter new salary : ");
76. salary = Integer.parseInt(sin.nextLine());
77. ;
78. e.set(pos, new Employee(empno, name, salary));
79. System.out.println("Employee updated successfully");
80. }
81. }
82. public static void main(String[] args) {
83. Scanner in = new Scanner(System.in);
84. int choice = -1;
85. List<Employee> employees = new ArrayList<>();
86. while (true) {
87. System.out.println("Enter 1 to insert");
88. System.out.println("Enter 2 to display");
89. System.out.println("Enter 3 to search");
90. System.out.println("Enter 4 to delete");
91. System.out.println("Enter 5 to update");
92. System.out.println("Enter 0 to exit");
93. System.out.print("Enter choice : ");
94. choice = Integer.parseInt(in.nextLine());
95. ;
96. switch (choice) {
97. case 1:
98. insert(employees);
99. break;
100. case 2:
101. display(employees);
102. break;
103. case 3:
104. search(employees);
105. break;
106. case 4:
107. delete(employees);
108. break;
109. case 5:
110. update(employees);
111. break;
112. case 0:
113. System.exit(0);
114. default:
115. break;
116. }
117. }
118. }
119. }

Output :

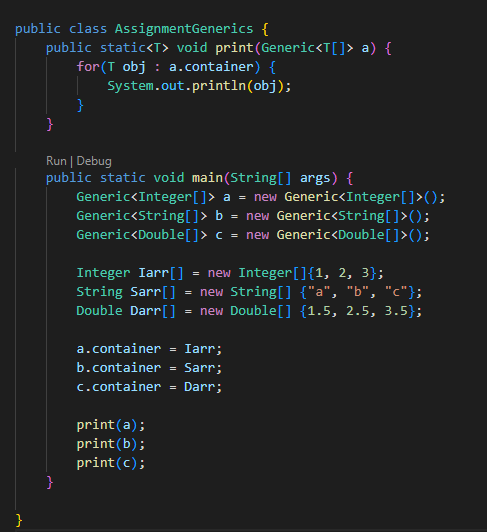




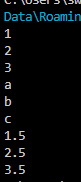
1. Generic Class



TesterClass



Output:



1. MyCalculator GUI Program

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class MyCalculator implements ActionListener {

    JFrame f;

    JTextField t;

    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, b0, badd, bsub, bdiv, bmul, bdec, beq, bdel, bclr;

    static double a = 0.0d, b = 0.0d, result = 0.0d;

    static String operator = "";

    public static boolean checkForOpeators(String s) {

        boolean ok = false;

        for (int i = 0; i < s.length(); i++) {

            if (s.charAt(i) == '+' || s.charAt(i) == '-' || s.charAt(i) == '\*' || s.charAt(i) == '/') {

                ok = true;

                break;

            }

        }

        return ok;

    }

    public static void calculateResult() {

        switch (operator) {

            case "+":

                result = a + b;

                break;

            case "-":

                result = a - b;

                break;

            case "\*":

                result = a \* b;

                break;

            case "/":

                result = a / b;

                break;

            default:

                result = a;

        }

    }

    public MyCalculator() {

        f = new JFrame("My Calculator");

        f.setVisible(true);

        f.setLayout(null);

        f.setBounds(20, 20, 300, 350);

        f.setResizable(false);

        Font fo = new Font("Arial", Font.BOLD, 20);

        t = new JTextField();

        t.setFont(fo);

        t.setBackground(Color.LIGHT\_GRAY);

        t.setForeground(Color.BLACK);

        t.setBounds(40, 40, 200, 50);

        b1 = new JButton("1");

        b2 = new JButton("2");

        b3 = new JButton("3");

        b4 = new JButton("4");

        b5 = new JButton("5");

        b6 = new JButton("6");

        b7 = new JButton("7");

        b8 = new JButton("8");

        b9 = new JButton("9");

        b0 = new JButton("0");

        badd = new JButton("+");

        bsub = new JButton("-");

        bdiv = new JButton("/");

        bmul = new JButton("\*");

        bdec = new JButton(".");

        beq = new JButton("=");

        bdel = new JButton("DEL");

        bclr = new JButton("CLR");

        b7.setBounds(40, 100, 50, 40);

        b8.setBounds(90, 100, 50, 40);

        b9.setBounds(140, 100, 50, 40);

        bdiv.setBounds(190, 100, 50, 40);

        b4.setBounds(40, 140, 50, 40);

        b5.setBounds(90, 140, 50, 40);

        b6.setBounds(140, 140, 50, 40);

        bmul.setBounds(190, 140, 50, 40);

        b1.setBounds(40, 180, 50, 40);

        b2.setBounds(90, 180, 50, 40);

        b3.setBounds(140, 180, 50, 40);

        bsub.setBounds(190, 180, 50, 40);

        b0.setBounds(40, 220, 50, 40);

        bdec.setBounds(90, 220, 50, 40);

        badd.setBounds(140, 220, 50, 40);

        beq.setBounds(190, 220, 50, 40);

        bdel.setBounds(40, 260, 100, 40);

        bclr.setBounds(140, 260, 100, 40);

        b1.addActionListener(this);

        b2.addActionListener(this);

        b3.addActionListener(this);

        b4.addActionListener(this);

        b5.addActionListener(this);

        b6.addActionListener(this);

        b7.addActionListener(this);

        b8.addActionListener(this);

        b9.addActionListener(this);

        b0.addActionListener(this);

        badd.addActionListener(this);

        bsub.addActionListener(this);

        bmul.addActionListener(this);

        bdiv.addActionListener(this);

        bdec.addActionListener(this);

        beq.addActionListener(this);

        bdel.addActionListener(this);

        bclr.addActionListener(this);

        f.add(t);

        f.add(b1);

        f.add(b2);

        f.add(b3);

        f.add(b4);

        f.add(b5);

        f.add(b6);

        f.add(b7);

        f.add(b8);

        f.add(b9);

        f.add(b0);

        f.add(badd);

        f.add(bsub);

        f.add(bmul);

        f.add(bdiv);

        f.add(bdec);

        f.add(beq);

        f.add(bdel);

        f.add(bclr);

    }

    public static void main(String[] args) {

        new MyCalculator();

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == b1) {

            t.setText(t.getText().concat("1"));

        }

        if (e.getSource() == b2) {

            t.setText(t.getText().concat("2"));

        }

        if (e.getSource() == b3) {

            t.setText(t.getText().concat("3"));

        }

        if (e.getSource() == b4) {

            t.setText(t.getText().concat("4"));

        }

        if (e.getSource() == b5) {

            t.setText(t.getText().concat("5"));

        }

        if (e.getSource() == b6) {

            t.setText(t.getText().concat("6"));

        }

        if (e.getSource() == b7) {

            t.setText(t.getText().concat("7"));

        }

        if (e.getSource() == b8) {

            t.setText(t.getText().concat("8"));

        }

        if (e.getSource() == b9) {

            t.setText(t.getText().concat("9"));

        }

        if (e.getSource() == b0) {

            t.setText(t.getText().concat("0"));

        }

        if (e.getSource() == bdec) {

            String curr = t.getText();

            String[] nums = curr.split("[-+\*/]");

            if (nums[nums.length - 1].indexOf('.') == -1) {

                t.setText(t.getText().concat("."));

            }

        }

        if (e.getSource() == bclr) {

            t.setText("");

        }

        if (e.getSource() == bdel) {

            String str = t.getText();

            t.setText(" ");

            for (int i = 0; i < str.length() - 1; i++) {

                t.setText((t.getText() + str.charAt(i)));

            }

        }

        if (e.getSource() == badd) {

            if (t.getText().length() == 0)

                return;

            if (!checkForOpeators(t.getText()))

                t.setText(t.getText().concat("+"));

        }

        if (e.getSource() == bsub) {

            if (t.getText().length() == 0)

                return;

            if (!checkForOpeators(t.getText()))

                t.setText(t.getText().concat("-"));

        }

        if (e.getSource() == bmul) {

            if (t.getText().length() == 0)

                return;

            if (!checkForOpeators(t.getText()))

                t.setText(t.getText().concat("\*"));

        }

        if (e.getSource() == bdiv) {

            if (t.getText().length() == 0)

                return;

            if (!checkForOpeators(t.getText()))

                t.setText(t.getText().concat("/"));

        }

        if (e.getSource() == beq) {

            String curr = t.getText();

            String[] nums = curr.split("[-+\*/]");

            operator = curr.replaceAll("[^-+\*/]", "");

            if (nums.length > 0) {

                a = Double.parseDouble(nums[0]);

            }

            if (nums.length > 1) {

                b = Double.parseDouble(nums[1]);

            }

            calculateResult();

            t.setText(String.valueOf(result));

        }

    }

}

Output :

