Java Applet

Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Advantage of Applet

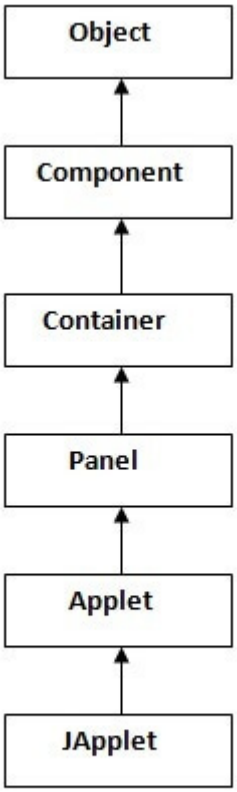
There are many advantages of applet. They are as follows:

- It works at client side so less response time.
- Secured
- It can be executed by browsers running under many plateforms, including Linux, Windows, Mac Os etc.

Drawback of Applet

- Plugin is required at client browser to execute applet.

Hierarchy of Applet



As displayed in the above diagram, Applet class extends Panel. Panel class extends Container which is the su

---

Lifecycle of Java Applet

1. Applet is initialized.
2. Applet is started.
3. Applet is painted.
4. Applet is stopped.
5. Applet is destroyed.

Lifecycle methods for Applet:

The java.applet.Applet class 4 life cycle methods and java.awt.Component class provides 1 life cycle methods for an applet.

java.applet.Applet class

For creating any applet java.applet.Applet class must be inherited. It provides 4 life cycle methods of applet.

1. **public void init():** is used to initialized the Applet. It is invoked only once.

2. **public void start():** is invoked after the init() method or browser is maximized. It is used to start the Applet.
3. **public void stop():** is used to stop the Applet. It is invoked when Applet is stop or browser is minimized.
4. **public void destroy():** is used to destroy the Applet. It is invoked only once.

Banking Application in Java

In this section, we will learn how to create a mini-application for a banking system in Java. In this program, we will add some basic functionalities of a bank account like a deposit of amount, withdrawal of amount, etc.

Initially, the program accepts the number of customers we need to add and adds the customer and account details accordingly. Further, it displays the series of menus to operate over the accounts.

The series of menus displayed are as follows:

1. Display all account details
2. Search by account number
3. Deposit the amount
4. Withdraw the amount
5. Exit

SOURCE CODE

```
1. import java.util.Scanner;
2. class BankDetails {
```

```
3.    private String accno;
4.    private String name;
5.    private String acc_type;
6.    private long balance;
7.    Scanner sc = new Scanner(System.in);
8.    //method to open new account
9.    public void openAccount() {
10.      System.out.print("Enter Account No: ");
11.      accno = sc.next();
12.      System.out.print("Enter Account type: ");
13.      acc_type = sc.next();
14.      System.out.print("Enter Name: ");
15.      name = sc.next();
16.      System.out.print("Enter Balance: ");
17.      balance = sc.nextLong();
18.   }
19.   //method to display account details
20.   public void showAccount() {
21.      System.out.println("Name of account holder: " + name);
22.      System.out.println("Account no.: " + accno);
23.      System.out.println("Account type: " + acc_type);
24.      System.out.println("Balance: " + balance);
25.   }
26.   //method to deposit money
27.   public void deposit() {
28.      long amt;
29.      System.out.println("Enter the amount you want to deposit: ");
30.      amt = sc.nextLong();
31.      balance = balance + amt;
32.   }
33.   //method to withdraw money
34.   public void withdrawal() {
35.      long amt;
36.      System.out.println("Enter the amount you want to withdraw: ");
37.      amt = sc.nextLong();
38.      if (balance >= amt) {
39.         balance = balance - amt;
```

```
40.            System.out.println("Balance after withdrawal: " + balance);
41.        } else {
42.            System.out.println("Your balance is less than " + amt + "\tTransaction failed...!
    !" );
43.        }
44.    }
45.    //method to search an account number
46.    public boolean search(String ac_no) {
47.        if (accno.equals(ac_no)) {
48.            showAccount();
49.            return (true);
50.        }
51.        return (false);
52.    }
53. }
54. public class BankingApp {
55.    public static void main(String arg[]) {
56.        Scanner sc = new Scanner(System.in);
57.        //create initial accounts
58.        System.out.print("How many number of customers do you want to input? ");
59.        int n = sc.nextInt();
60.        BankDetails C[] = new BankDetails[n];
61.        for (int i = 0; i < C.length; i++) {
62.            C[i] = new BankDetails();
63.            C[i].openAccount();
64.        }
65.        // loop runs until number 5 is not pressed to exit
66.        int ch;
67.        do {
68.            System.out.println("\n ***Banking System Application***");
69.            System.out.println("1. Display all account details \n 2. Search by Account num
    ber\n 3. Deposit the amount \n 4. Withdraw the amount \n 5.Exit ");
70.            System.out.println("Enter your choice: ");
71.            ch = sc.nextInt();
72.               switch (ch) {
73.                  case 1:
74.                     for (int i = 0; i < C.length; i++) {
```

```
75.                     C[i].showAccount();
76.                 }
77.                 break;
78.             case 2:
79.                 System.out.print("Enter account no. you want to search: ");
80.                 String ac_no = sc.next();
81.                 boolean found = false;
82.                 for (int i = 0; i < C.length; i++) {
83.                     found = C[i].search(ac_no);
84.                     if (found) {
85.                         break;
86.                     }
87.                 }
88.                 if (!found) {
89.                     System.out.println("Search failed! Account doesn't exist..!!");
90.                 }
91.                 break;
92.             case 3:
93.                 System.out.print("Enter Account no. : ");
94.                 ac_no = sc.next();
95.                 found = false;
96.                 for (int i = 0; i < C.length; i++) {
97.                     found = C[i].search(ac_no);
98.                     if (found) {
99.                         C[i].deposit();
100.                             break;
101.                         }
102.                     }
103.                     if (!found) {
104.                         System.out.println("Search failed! Account doesn't exist..!!");

105.                     }
106.                     break;
107.                 case 4:
108.                     System.out.print("Enter Account No : ");
109.                     ac_no = sc.next();
110.                     found = false;
```

```
111.                     for (int i = 0; i < C.length; i++) {
112.                         found = C[i].search(ac_no);
113.                         if (found) {
114.                             C[i].withdrawal();
115.                             break;
116.                         }
117.                     }
118.                     if (!found) {
119.                         System.out.println("Search failed! Account doesn't exist..!!");

120.                     }
121.                     break;
122.                 case 5:
123.                     System.out.println("See you soon...");
124.                     break;
125.                 }
126.             }
127.             while (ch != 5);
128.         }
129.     }
```

Multiple choice questions

1. Which of these functions is called to display the output of an applet?
   a) display()
   b) paint()
   c) displayApplet()
   d) PrintApplet()

2. Which of these methods can be used to output a string in an applet?
   a) display()
   b) print()
   c) drawString()
   d) transient()

3. Which of these methods is a part of Abstract Window Toolkit (AWT) ?
   a) display()
   b) paint()
   c) drawString()
   d) transient()

4. Which of these modifiers can be used for a variable so that it can be accessed from any thread or parts of a program?
   a) transient
   b) volatile
   c) global
   d) No modifier is needed

5. Which of these operators can be used to get run time information about an object?
   a) getInfo
   b) Info
   c) instanceof
   d) getinfoof