

Day 8

Interface

- Abstract helper class, which allows us to override some of the methods of interface is called adapter class.
- It is frequently used in AWT, Servlet etc.

```
interface A
{
    void f1();
    void f2();
    void f3();
}
abstract class B implements A    //Adapter class
{
    @Override
    public void f1() {        }
    @Override
    public void f2() {        }
    @Override
    public void f3() {        }
}
class C extends B
{
    @Override
    public void f2()
    {
        System.out.println("C.f2");
    }
}
public class Program
{
    public static void main(String[] args)
    {
        A a = new C( );
        a.f2( );
    }
}
```

Default method

- If we want to modify definition of interface at runtime then we should use default method.
- It is mandatory to provide body for default method.

- Overriding default method is optional.
- If we want call default method in sub class then we should use following syntax:
 InterfaceName.super.defaultMethodName();
- If interface having default method with same name then overrrding default method is mandatory.

```
interface A
{
    default void f1( )
    {
        System.out.println("A.f1");
    }
    default void f3( )
    {
        System.out.println("A.f3");
    }
}
interface B
{
    default void f2( )
    {
        System.out.println("B.f2");
    }
    default void f3( )
    {
        System.out.println("B.f3");
    }
}
class C implements A, B
{
    @Override
    public void f3() //mandatory to override
    {
        System.out.println("C.f3");
        A.super.f3();
        B.super.f3();
    }
}
```

- Interface static method is called utility method.
- Such methods are not designed to override rather its is designed to support default methods.

Comparable

- It is interface declared in java.lang package.
- "int compareTo(T other)" is a method of Comparable interface.
- If we want to sort, array of instances of reference type which contains all the elements of same type then reference type must implement Comparable interface.
- "compareTo" method returns integer value(-1, 0, 1)
 1. If state of current object is less then it should return any negative value(Generally it is -1)
 2. If state of current object is equal then it should return zero(0).
 3. If state of current object is greater then it should return any positive value(Generally it is 1)

Comparator

- It is interface declared in java.util package.
- "int compare(T o1,T o2)" is a method of comparator interface.
- If we want to sort, array of instances of reference type which contains all the elements of different type then reference type must implement Comparator interface.
- "compare" method returns integer value(-1, 0, 1)
 1. If state of current object is less then it should return any negative value(Generally it is -1)
 2. If state of current object is equal then it should return zero(0).
 3. If state of current object is greater then it should return any positive value(Generally it is 1)