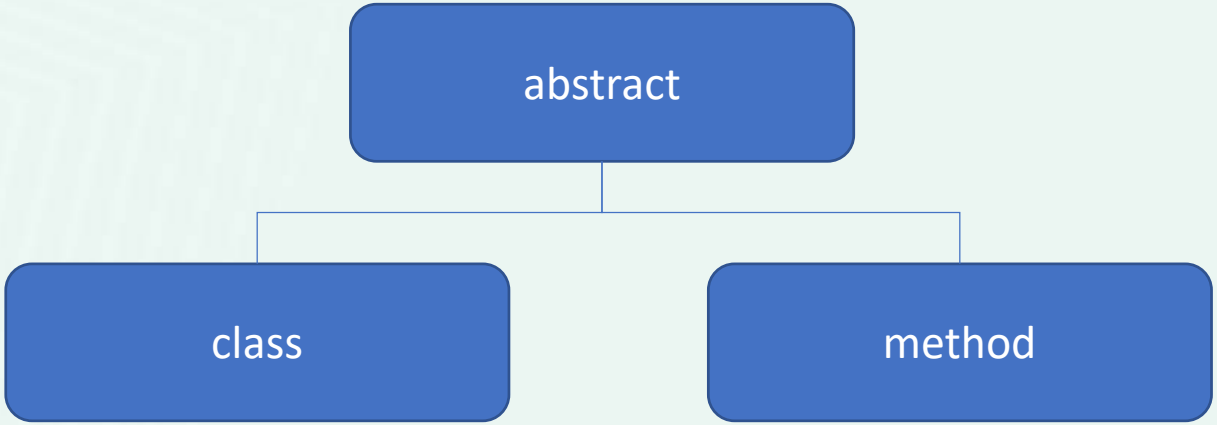
Abstraction. Interface

Abstraction – bu o’zbek tilida mavhumlik degan ma’noni beradi. Dasturlashda umumlashtirish maqsadida ishlatiladi.


abstract
class
method

Abstract qilib e’lon qilingan class abstract class deyiladi.

Abstract classdan object olib bo’lmaydi.

```
public abstract class Animal {

}
```

Abstract deb e’lon qilingan va body qismi bo’lmagan method abstract method deyiladi.

Abstract methodni abstract bo’lmagan class ichida e’lon qilib bo’lmaydi.

```
public abstract void voice();
```

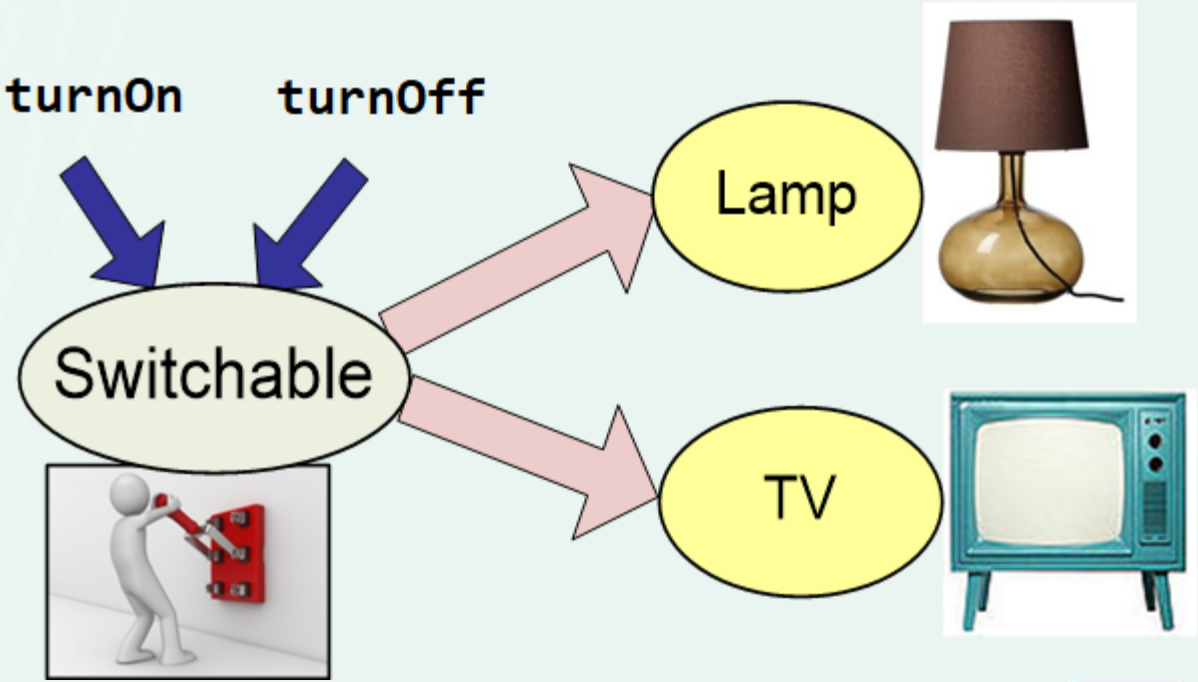
Abstract class ichida constructor, o’zgaruvchi hamda abstract va abstract bo’lmagan methodlar bo’lishi mumkin.

```
public abstract class Animal {

        int age=1;
        public static double NUMBER_OF_EYE=2;

        public abstract void voice();

        public void about(){
              System.out.println("I'm animal");
        }
}
```

Interface

Interface –bu ko’pchilik obektlar yoki holatlarda mavjud bo’lgan funksionallikdir.

Interface – bu klassning boshqa klasslar foydalanishi mumkin bo’lgan methodlar jamlamasidir

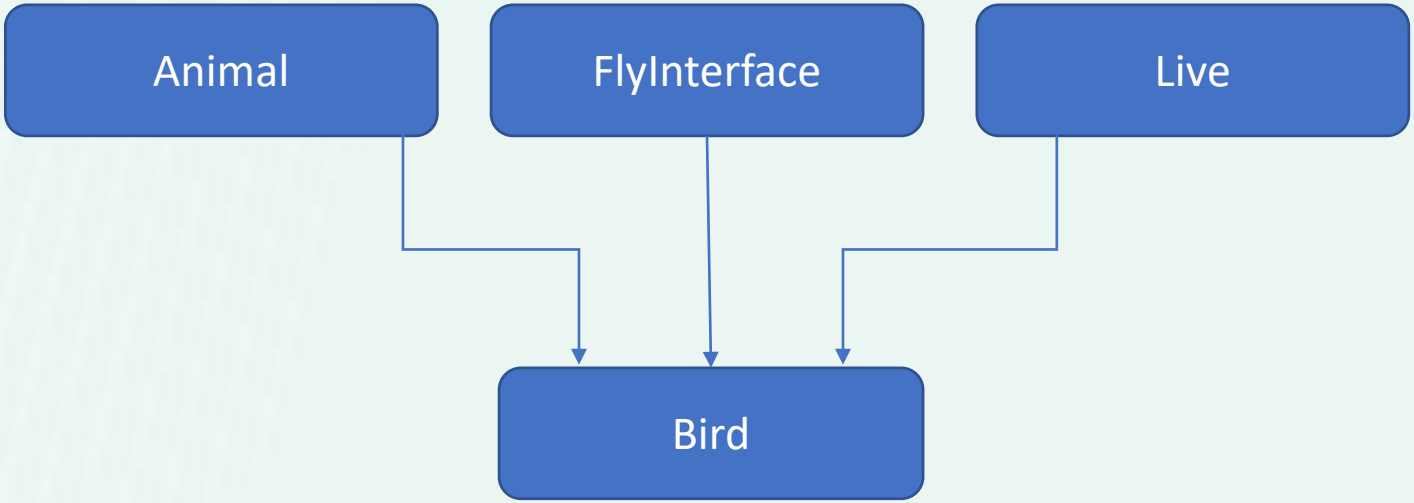

turnOn
turnOff
Switchable
Lamp
TV

```
public interface Technical {

        void turnOn();
        void turnOff();
        void repair();

}
```

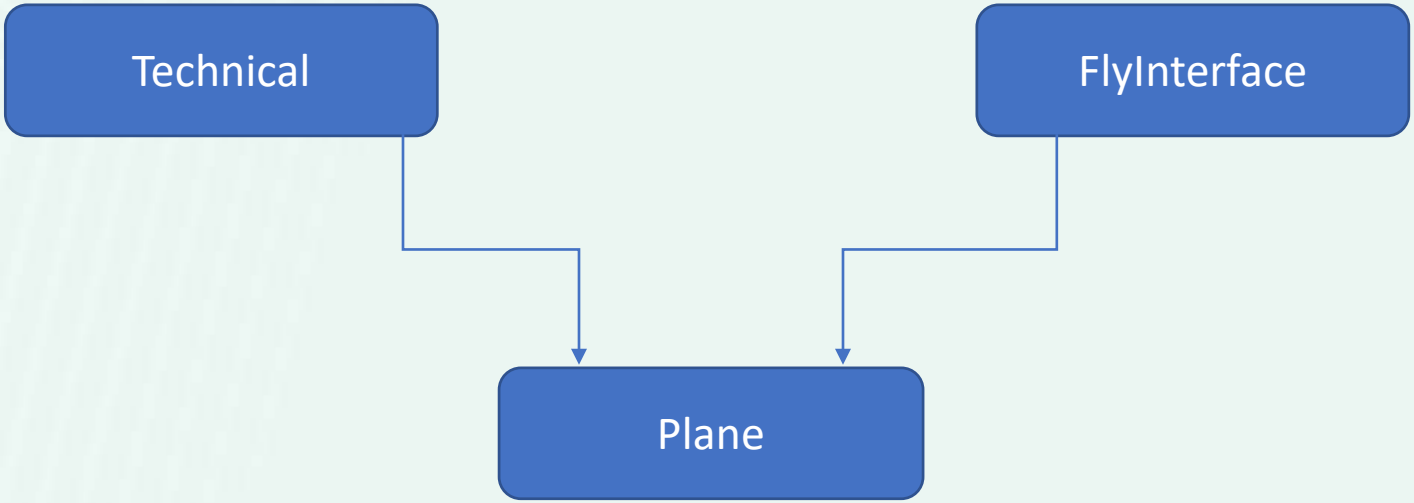
```java
public interface LiveInterface {

        void eat();
        void sleep();
        void walk();

}
```

```
public interface FlyInterface {

        void fly();
        void landing();

}
```

Multiple Inheritance



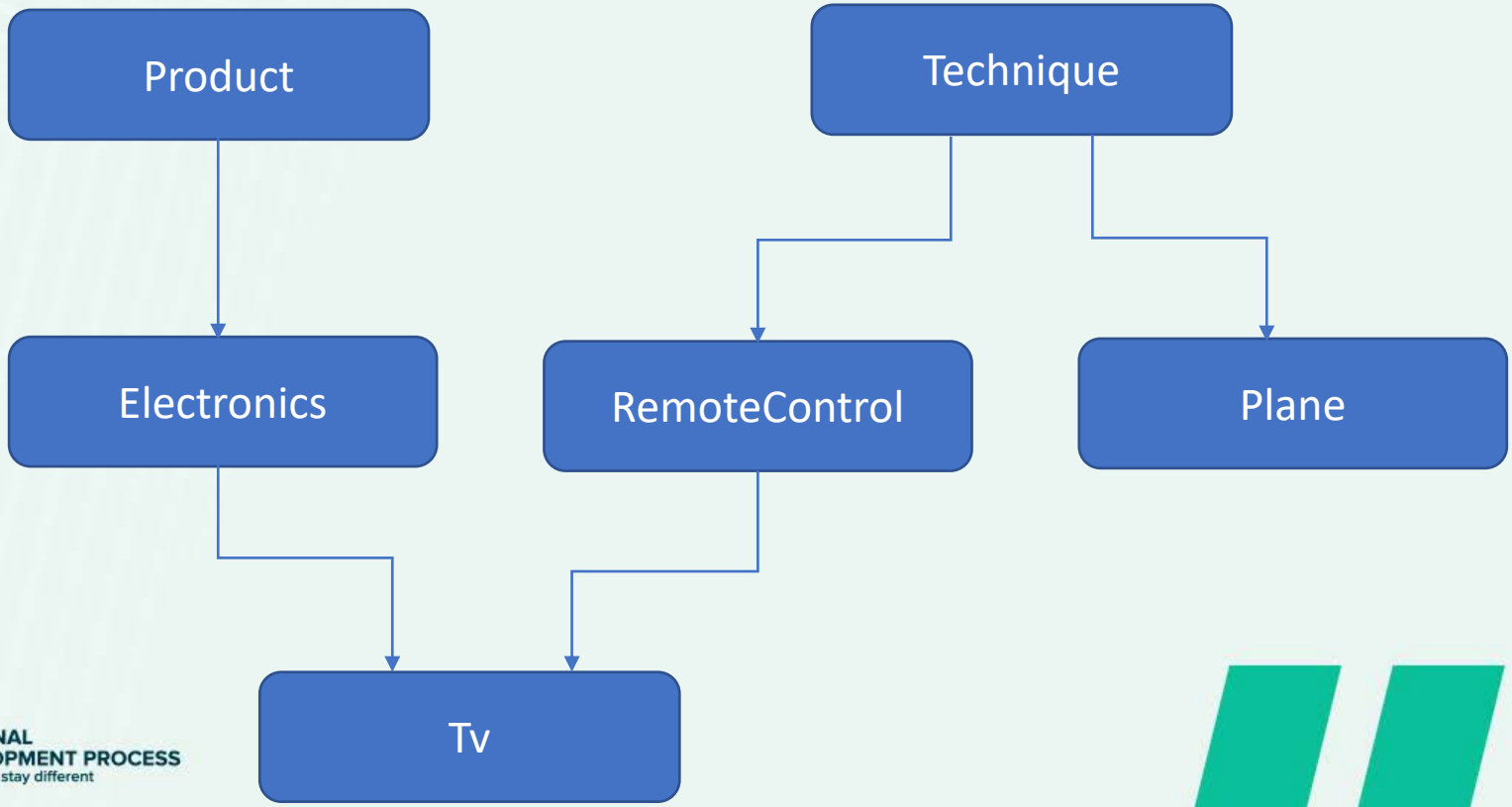
```java
public class Bird extends Animal implements FlyInterface,Live {
    private String type;
    private String color;
    private double weight;

    @Override
    public void voice() {

    }

    @Override
    public void fly() {

    }

    @Override
    public void landing() {

    }

    @Override
    public void eat() {

    }

    @Override
    public void walk() {

    }

//    getters and setters
}
```


Technical
FlyInterface
Plane

```java
public class Plane implements Technical, FlyInterface {
    private String model;
    private int capacity;
    private int numberOfEngine;

    @Override
    public void fly() {

    }

    @Override
    public void landing() {

    }

    @Override
    public void turnOn() {

    }

    @Override
    public void turnOff() {

    }

    @Override
    public void repair() {

    }

    //    getters and setters
}
```

```java
public class Tv extends Electronics implements RemoteControl {
    @Override
    public void changeChannel() {

    }

    @Override
    public void editSettings() {

    }

    @Override
    public void turnOn() {

    }

    @Override
    public void turnOff() {

    }

    @Override
    public void repair() {

    }
}
```

Hybrid Inheritance



Abstract class vs Interface

- Abstract class dan voris olish ***extends***, interfacedan esa ***impliments*** kalit so'zi o'rqali amalga oshiriladi.
- Abstract classni bir-biriga chambarchas bo'g'liq bo'lgan classlarga nisbatandan qo'llanilsa, interfaceni bir-biriga bog'liq bo'lmagan class yoki interfacelarga nisbatan ham qo'llasa bo'ladi.
- Abstract classni ichida abstract bo'lmagan methodni e'lon qilish mumkin, interfaceda mumkin emas.
- Abstract class da oddiy o'zgaruvchi e'lon qilish mumkin, interface da e'lon qilingan oddiy o'zgaruvchi avtomatik tarzda ***public static final*** bo'ladi.
- Abstract class IS-A (shu toifa(class)ga mansubmi) savoliga, interface HAS-A (shunaqa xoss(method)aga egami) savoliga javob bo'ladi.
- Interfaceda barcha method va o'zgaruvchilar public (yozilmagan bo'lsa ham) bo'lib elon qilinadi

Key points: Here are the key points to remember about interfaces:

1) We can’t instantiate an interface in java. That means we cannot create the object of an interface
2) Interface provides full abstraction as none of its methods have body. On the other hand abstract class provides partial abstraction as it can have abstract and concrete(methods with body) methods both.
3) implements keyword is used by classes to implement an interface.
4) While providing implementation in class of any method of an interface, it needs to be mentioned as public.
5) Class that implements any interface must implement all the methods of that interface, else the class should be declared abstract.
6) Interface cannot be declared as private, protected or transient.
7) All the interface methods are by default **abstract and public**.
8) Variables declared in interface are **public, static and final** by default.
9) Interface variables must be initialized at the time of declaration otherwise compiler will throw an error.
10) Inside any implementation class, you cannot change the variables declared in interface because by default, they are public, static and final. Here we are implementing the interface “Try” which has a variable x. When we tried to set the value for variable x we got compilation error as the variable x is public static final by default and final variables can not be re-initialized.

- An interface can extend any interface but cannot implement it. Class implements interface and interface extends interface.
- 12) A class can implement any number of interfaces.
- 13) If there are two or more same methods in two interfaces and a class implements both interfaces, implementation of the method once is enough.

14) A class cannot implement two interfaces that have methods with same name but different return type.
15) Variable names conflicts can be resolved by interface name.