Day 4 : Interfaces, Abstract classes and static

Abstraction

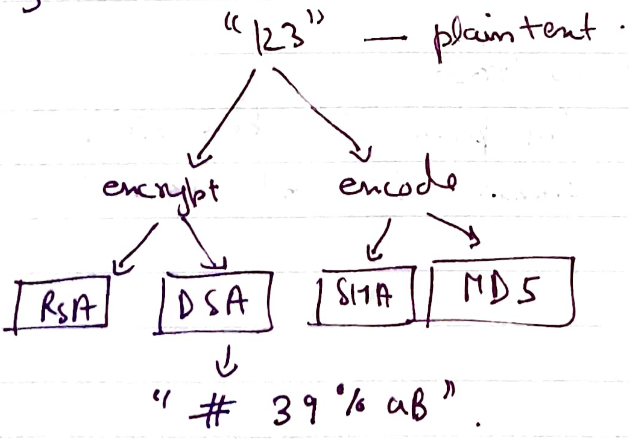
1. Encapsulation.
   1. classes — single unit.
   2. access modifiers — Info hiding.

2. Inheritance.
   1. Reusability — extends.

3. Polymorphism.
   1. Generic interfaces — subtyping.
   2. Dev. experience — overloading.
   3. Diff. implemantions — overriding.

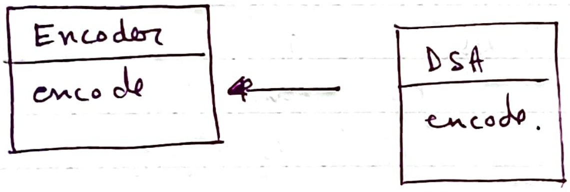
1. Interfaces.
2. Abstract classes.
3. static

```
class User {
      string email;
      String username;
      String password;
}
```



```
class Password Encoder {

    encode () {
        throw an error()
    }

}

class RSA Encoder extends PE {

      @override.
      encode () {
          RSA
      }
}.
```

contract — class # encode.



① Interface

- ↳ contract
- ↳ Blue print for behaviour
- ↳ cannot instantiate.

```
Interface Encoder {
    public string encode (string pw);
}
```

↓

Method declarations.

```
class RSA implements Encoder {
    public string encode ( string pw) {
        ... RSA .
    }
    public string ↓ internal Encode ()
}
```

RSA {	DSA {	MD5 { X
• encode	• encode pw	• toencoded
}	}	}
RSA ← PE	DSA ← PE	MD5 ← PE
encode	encode	encode

List < Password Encoder> . encode ()

List = ArrayList | Linked List .

Map = HashMap .

Code to an interface

Java 8 >

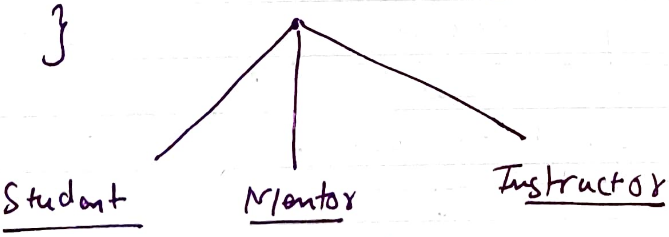
① default

```
default encode () {
    . . . . . . . .
}
```

```
interface Decoder {
    decoder ();
}.
```

```
② class RSA implements Encoder, Decoder {
}.
```

```
User {
    String email; ——O
    public login () { }
}
```



→ overridden.

① class
- → Fields .
- → methods
  - → implemented → get Email.
  - → not implemented
    - ↳ Login

Abstract
- ↳ not clean .
- ↳ incomplete .

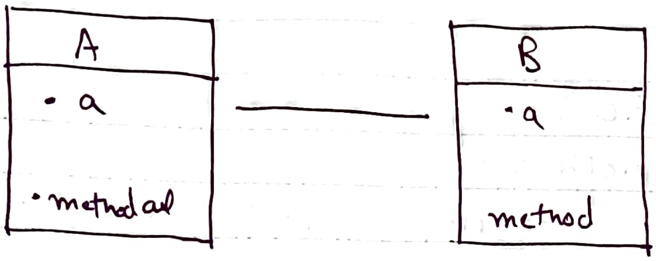
Abstract classes
- ↳ not instantiatoble .
- ↳ static + behaviour .
  - impl.
  - abstract

abstract class

↑

①

```
Password Encoder {
    string publicKey ;          → state .

  public (abstract) string encode () ;      → Non-imb.

  public string get key () {
      return publicKey ;
  }                                          ↓ Imp methods
}
```

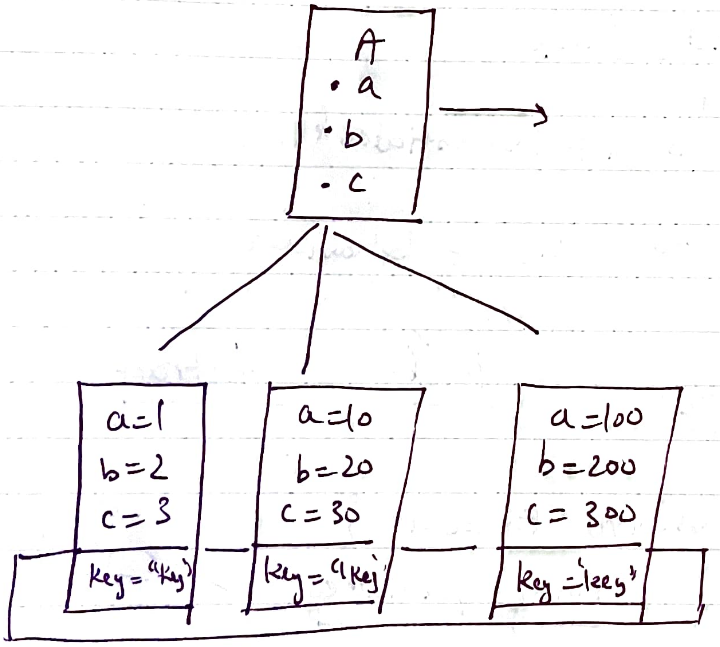


① No common attributes?

→ $\boxed{\text{Interface}}$

② If yes,

→ Abstract.



→ int → 4bytes

→ 36 bytes

8 bytes

+ 3 × 8

= $\boxed{\text{60 bytes}}$

44

Static

↳ memory management

Variable → Instance

↓

Class variable.

↳ shared across instances

```
private string a;
Private static string a;
```

```
User {
    name
     getName () →
Static   print () {
        sysout ("Hello");
    }
}
```

```
User   u = new   User();
   u.getName();
```

vs

```
User.print();
```

Static

Static — non - Static

```
Print () {
    getName () →
}
```

Static → Static ✓

non - static → static ✓