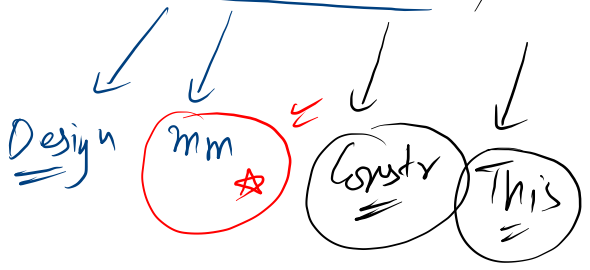
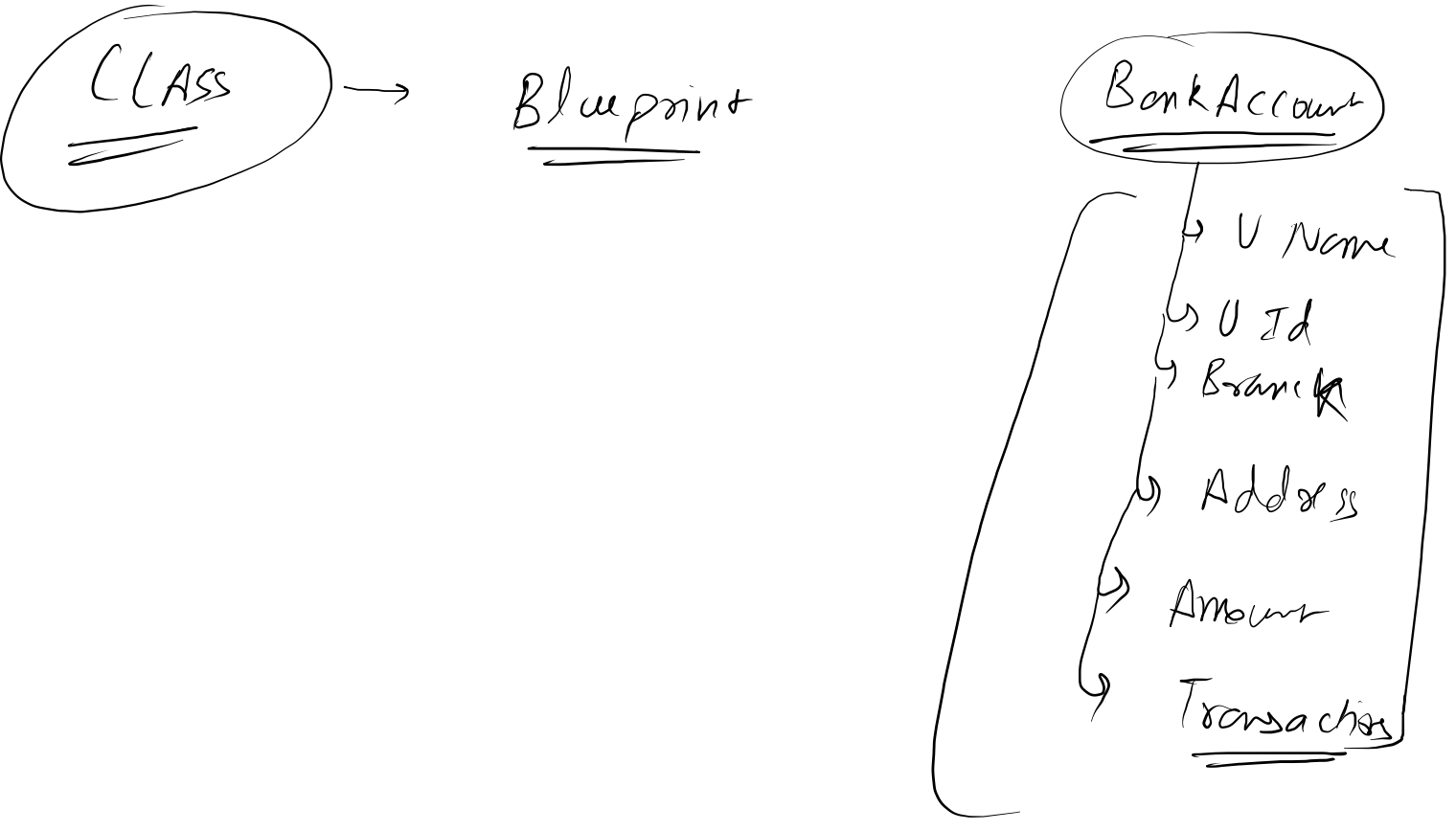
OOP [(Classes & objects)]



(Samsung → Stark → Gusha)

Java

- → Inheritance
- → oop (Class, mm)
- → Error handling

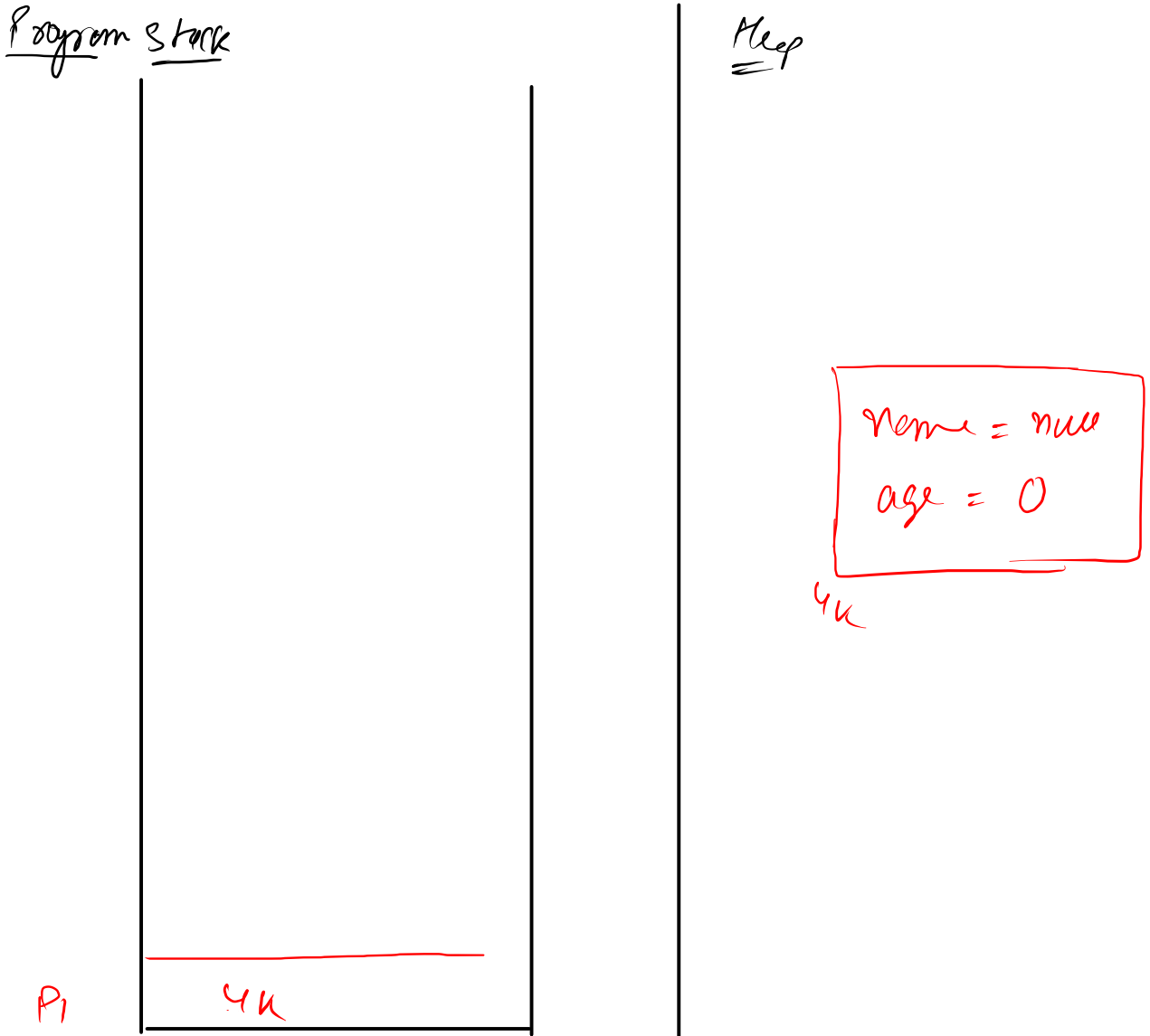

CLASS
Blueprint
BankAccount
U Name
U Id
Branch
Address
Amount
Transaction

Blueprint

```
public static class Person{

    String name;
    int age

}
```

```
Person p1 = new Person();
```

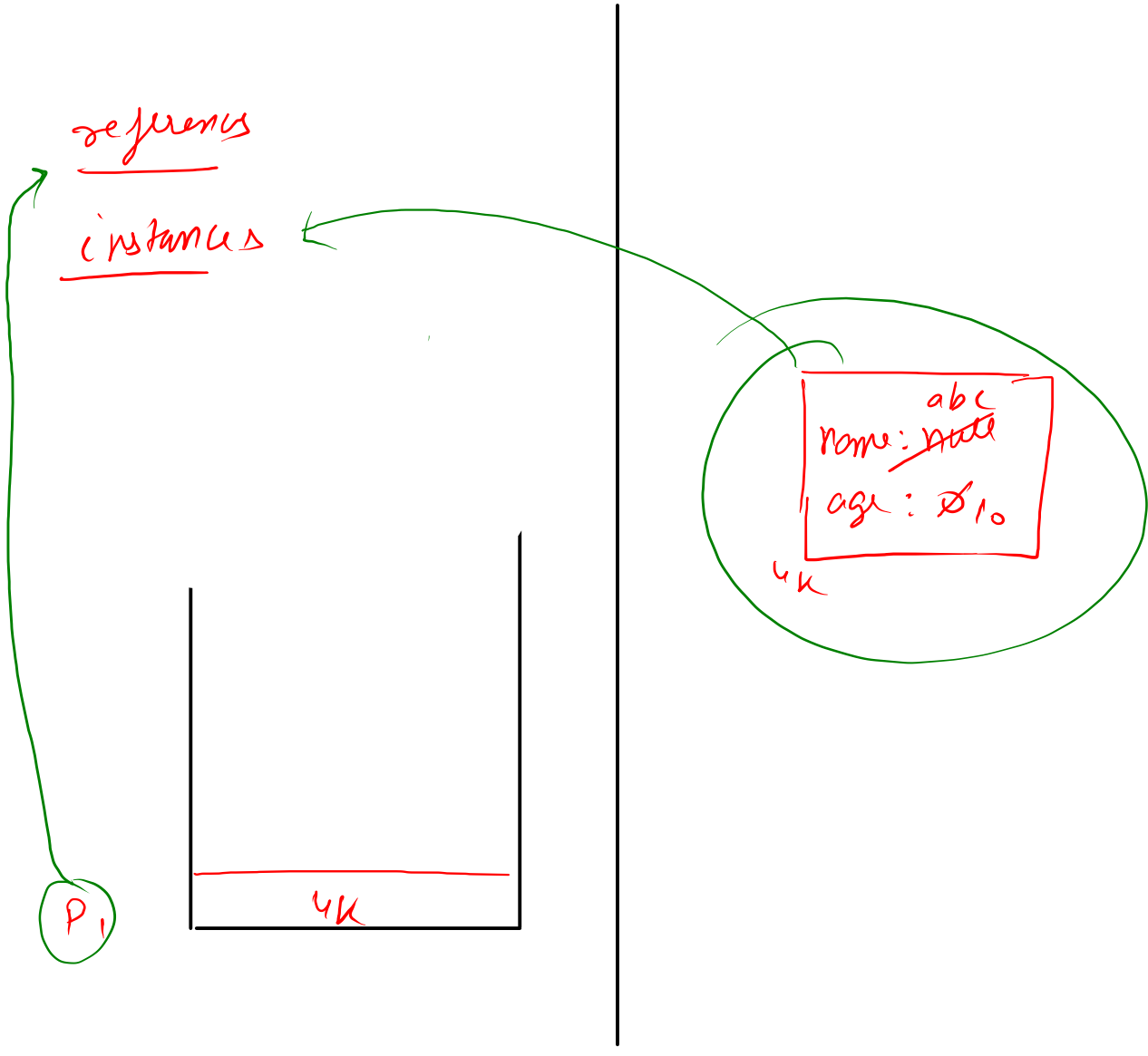
```java
public static class Person{
    String name;
    int age;
}

Run | Debug
public static void main(String[] args) {
    Person p1 = new Person();
}
```



```java
public static class Person{
    String name;
    int age;

    public String toString(){
        return name+" --> "+age;
    }

    public void saysHi(){
        System.out.println(name + " ["+age+"] says hi");
    }
}

Run | Debug
public static void main(String[] args) {
    Person p1 = new Person();

    // System.out.println(p1.name+" --> "+p1.age);

    // System.out.println(p1);
    p1.name = "abc";
    p1.age = 10;
    p1.saysHi();
}
```

abc [10] says hi;

Class → data members + member functions

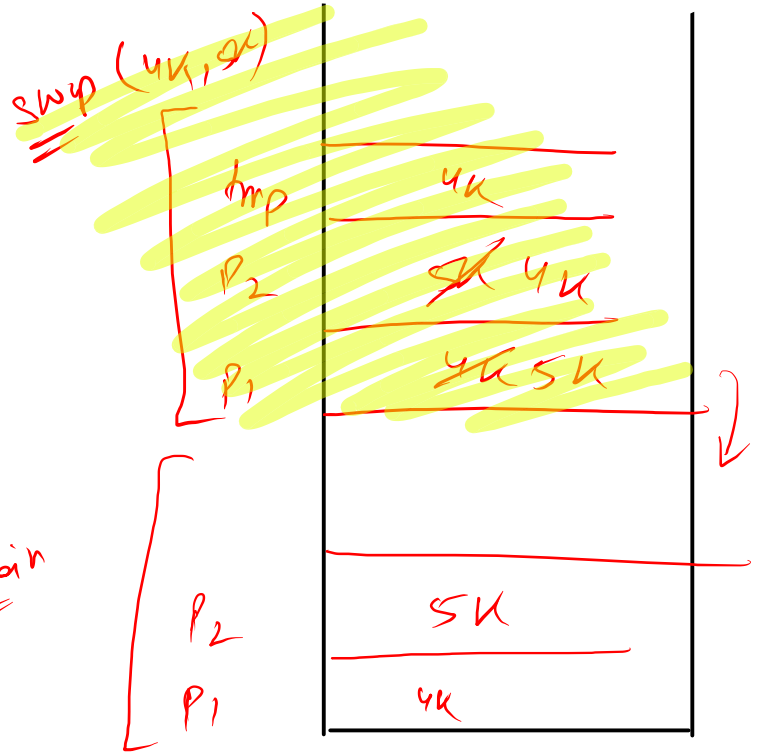


```java
public static void swap(Person p1 , Person p2){
    Person tmp = p1;
    p1 = p2;
    p2 = tmp;
}
Run | Debug
public static void main(String[] args) {

    Person p1 = new Person();
    p1.name = "abc";
    p1.age = 10;
    Person p2 = new Person();
    p2.name = "def";
    p2.age = 20;

    p1.saysHi();
    p2.saysHi();
    swap(p1, p2);
    p1.saysHi();
    p2.saysHi();
```

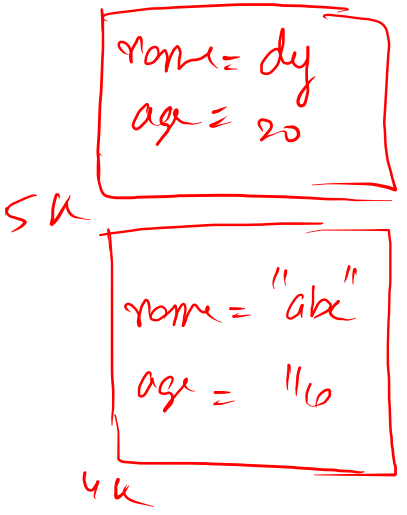
abc[10]
dy[20]
abc[10]
dy[20]

Program Stack

No Swapping

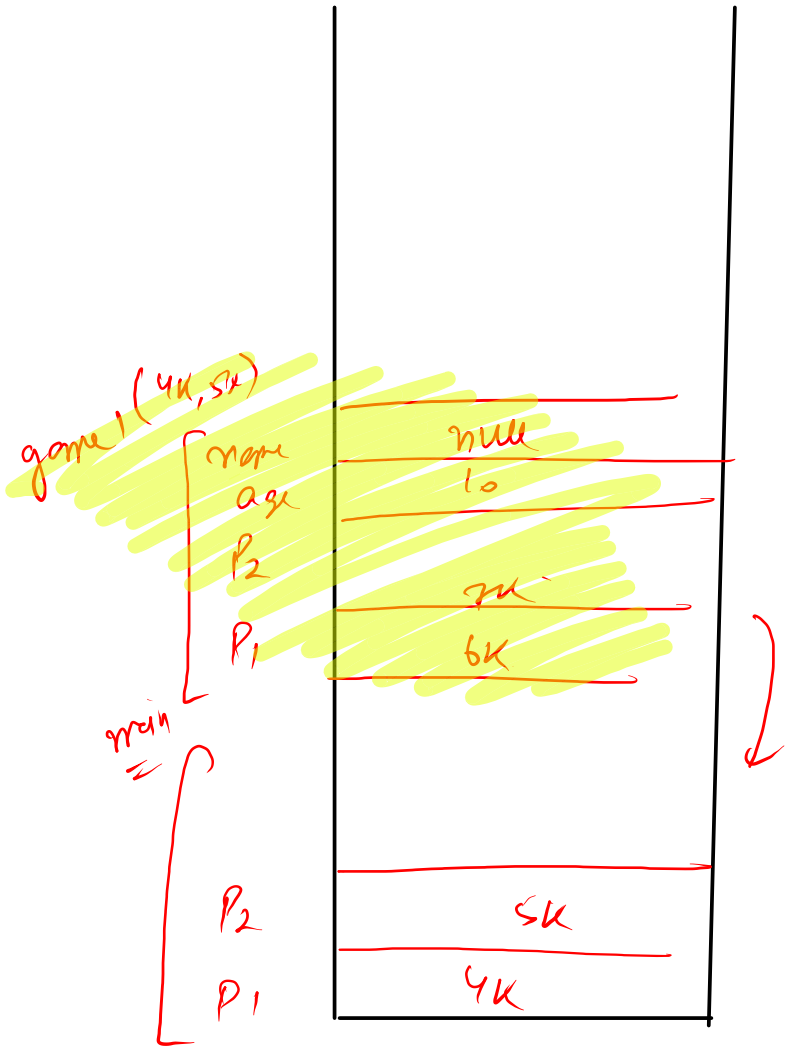


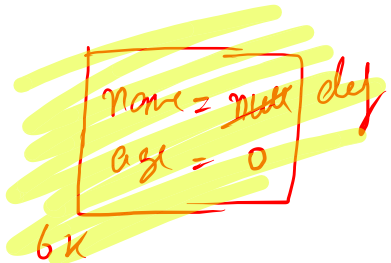
Heap

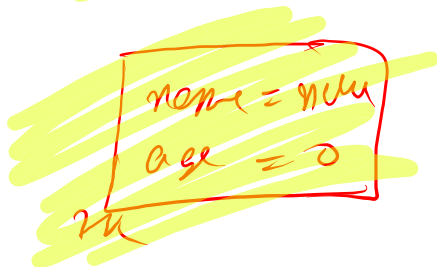


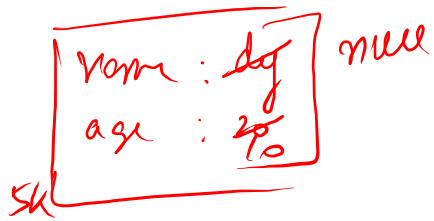
```java
public static void game1(Person p1,Person p2){
    int age = p1.age;
    p1.age = p2.age;
    p2.age = age;
    p1 = new Person();

    String name = p1.name;

    p1.name = p2.name;
    p2.name = name;
    p2 = new Person();
}
Run | Debug
public static void main(String[] args) {

    Person p1 = new Person();
    p1.name = "abc";
    p1.age = 10;
    Person p2 = new Person();
    p2.name = "def";
    p2.age = 20;

    p1.saysHi();
    p2.saysHi();
    game1(p1, p2);
    p1.saysHi();
    p2.saysHi();
```

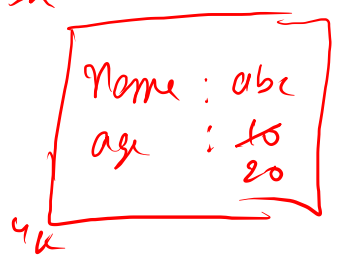
abc [10]
def [20]
abc [20]
null [10]



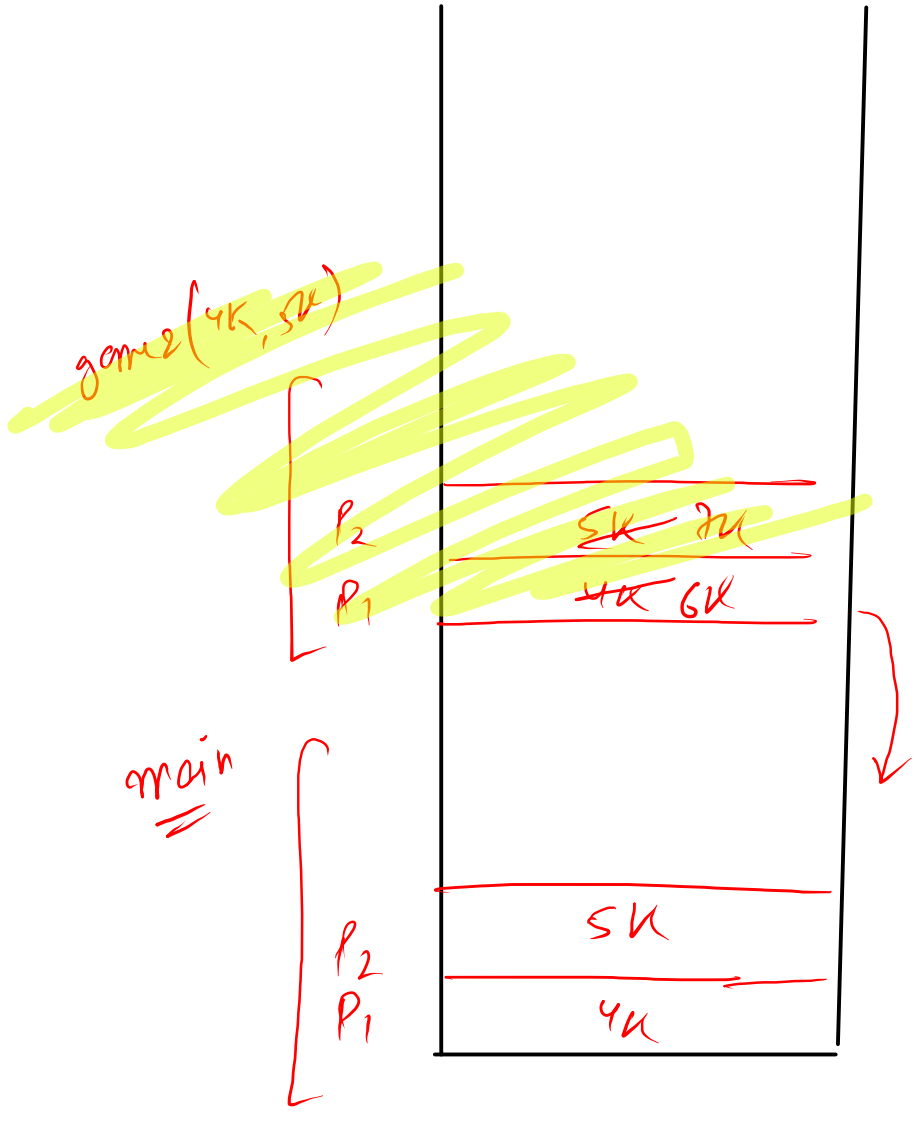


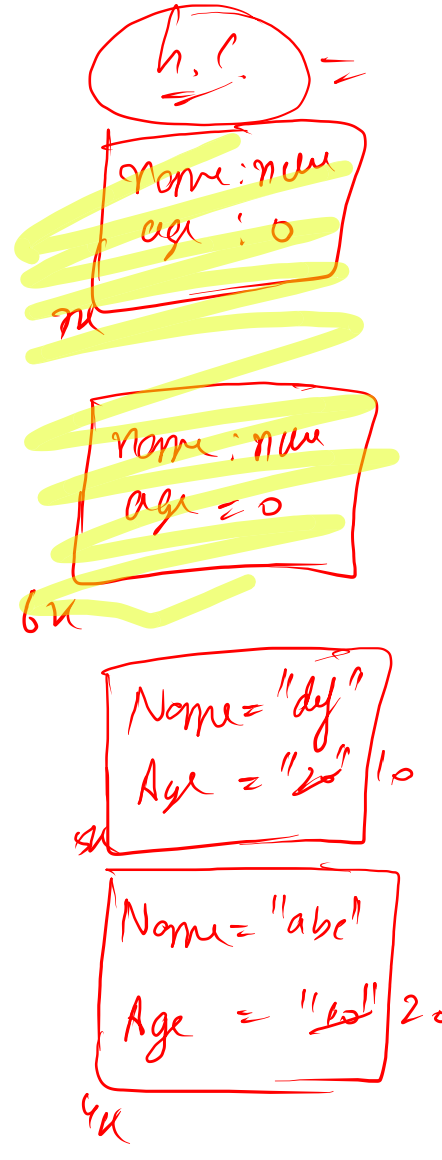






```java
public static void game2(Person p1,Person p2){
    int age = p1.age;
    p1.age = p2.age;
    p2.age = age;

    p1 = new Person();
    p2 = new Person();

    String name = p1.name;
    p1.name = p2.name;
    p2.name = name;
}
Run | Debug
public static void main(String[] args) {

    Person p1 = new Person();
    p1.name = "abc";
    p1.age = 10;
    Person p2 = new Person();
    p2.name = "def";
    p2.age = 20;

    p1.saysHi();
    p2.saysHi();
    game2(p1, p2);
    p1.saysHi();
    p2.saysHi();
```

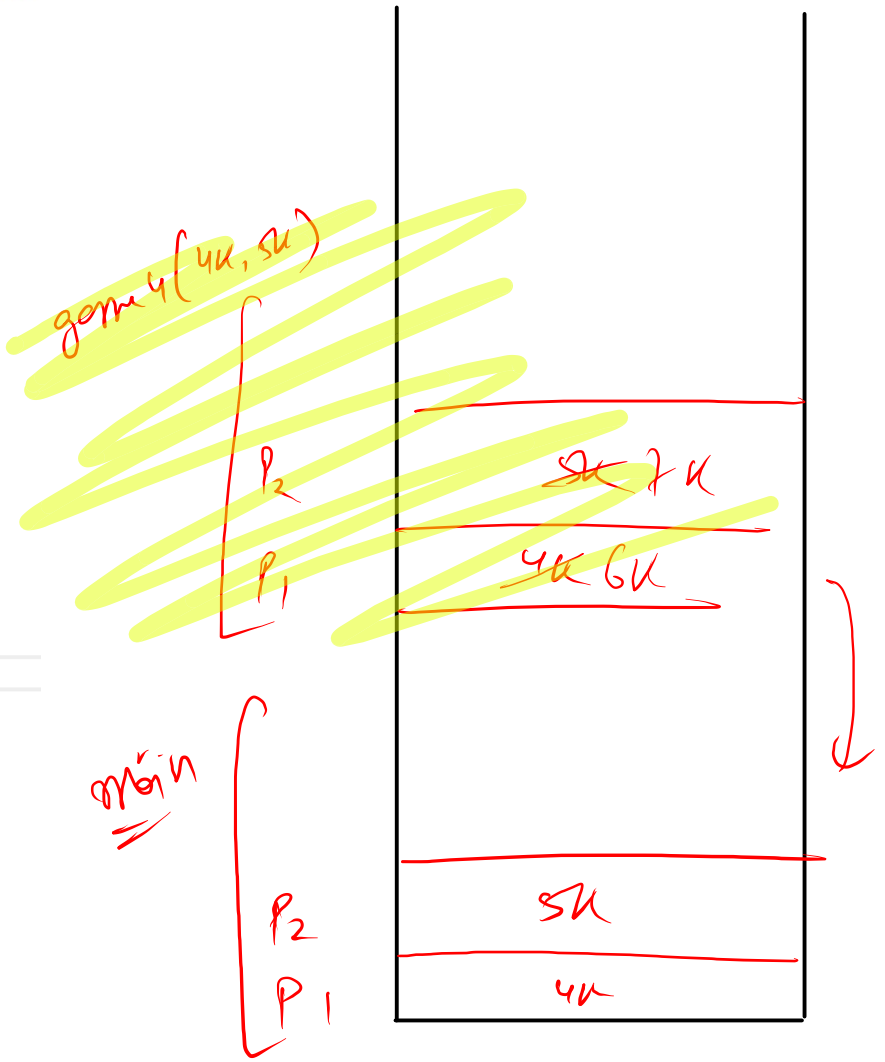


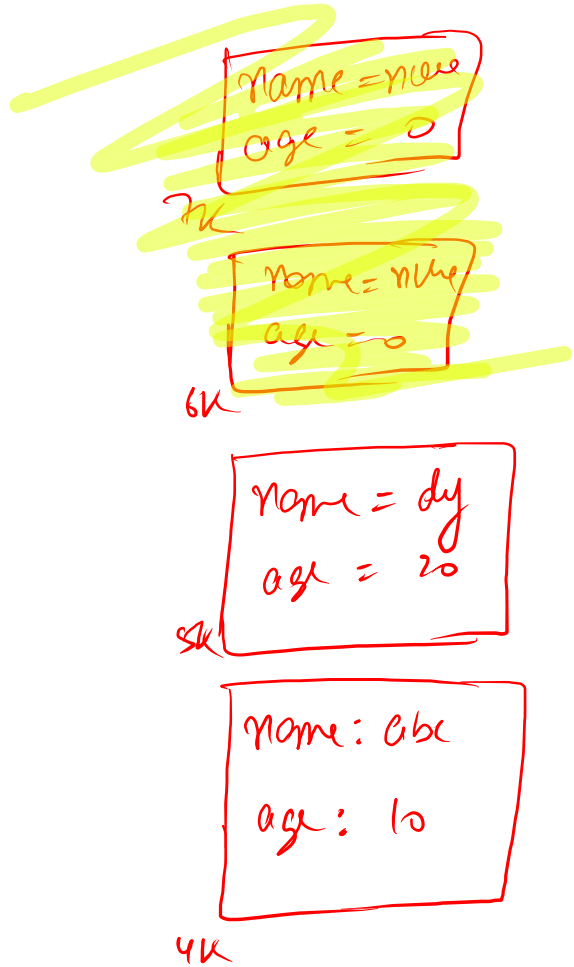


```java
public static void game4(Person p1,Person p2) {
    p1 = new Person();
    p2 = new Person();

    int age = p1.age;
    p1.age = p2.age;
    p2.age = age;


    String name = p1.name;
    p1.name = p2.name;
    p2.name = name;

}
Run | Debug
public static void main(String[] args) {

    Person p1 = new Person();
    p1.name = "abc";
    p1.age = 10;
    Person p2 = new Person();
    p2.name = "def";
    p2.age = 20;

    p1.saysHi();
    p2.saysHi();
    game4(p1, p2);
    p1.saysHi();
    p2.saysHi();
```

6K, 7K

p1.saysHi(); abc [10]
p2.saysHi(); def [20]
p1.saysHi(); → abc [10]
p2.saysHi(); → dy [20]





Constructors ☆

↳ default

↳ Parameterized

☆ =

↳ Java → default blank Constructor

Constructors

- same as class name
- no return type
- automatically call [instance create]
  - new

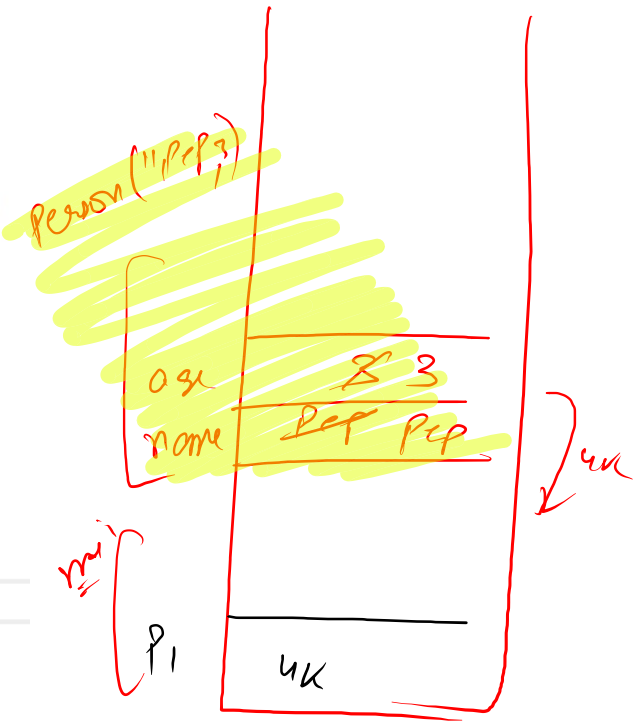
Uses

- msg
- warning
- default values
  - string → value → sample
  - age → 0 → 1
- Preprocessing
- at the beggin

* Default constructor is provided by java only if no constructor is defined by user

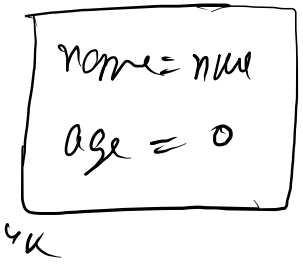
data member

```
    Person(String name,int age){
        System.out.println("paramterised constructor");
        name = name;
        age = age;
    }
}

Run | Debug
public static void main(String[] args) {
    Person p1 = new Person("pep",3);
    p1.saysHi();
```



Parameterized constructor

null[0]



this → self referenced

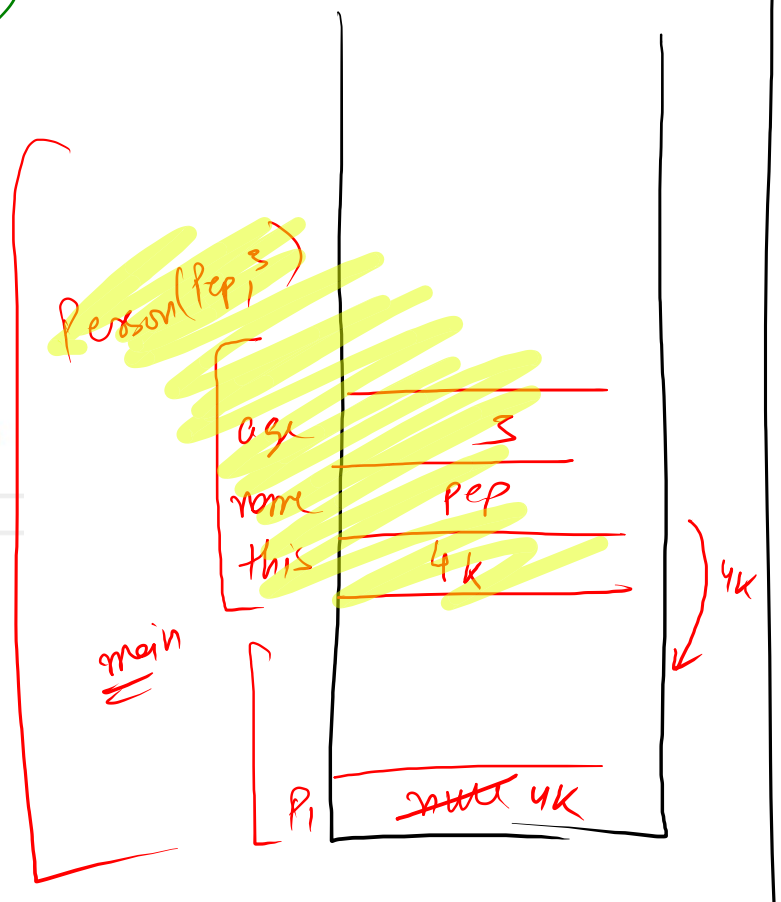
+

member functions

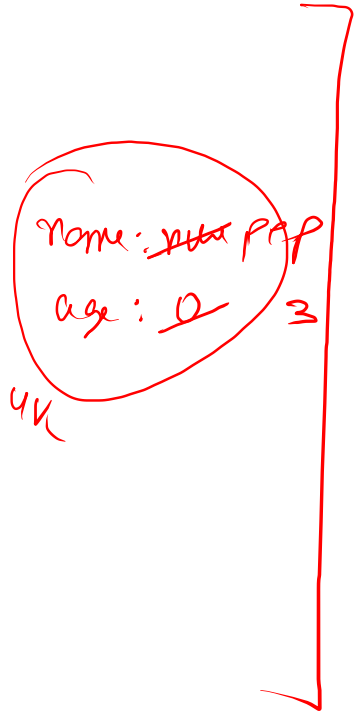
data members

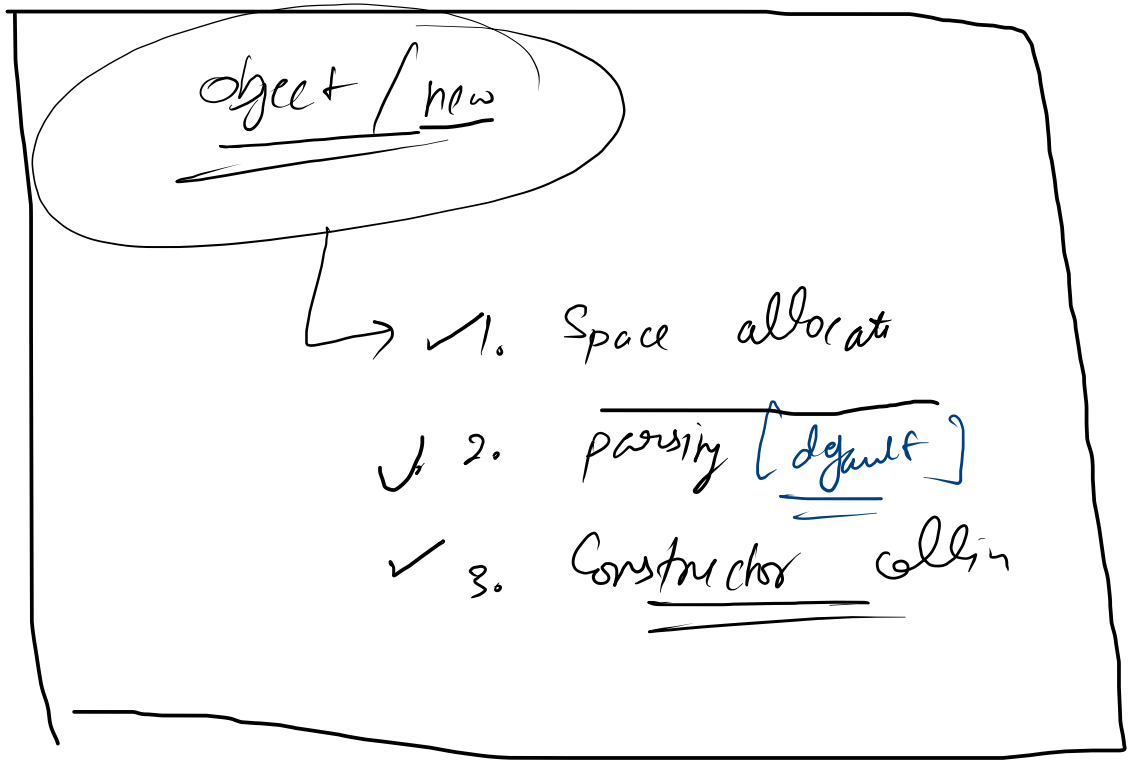
This

```
    Person(String name,int age){
        System.out.println("paramterised constructor");
        this.name = name;
        this.age = age;
    }
}

Run | Debug
public static void main(String[] args) {
    Person p1 = new Person("pep",3);
    p1.saysHi();
```

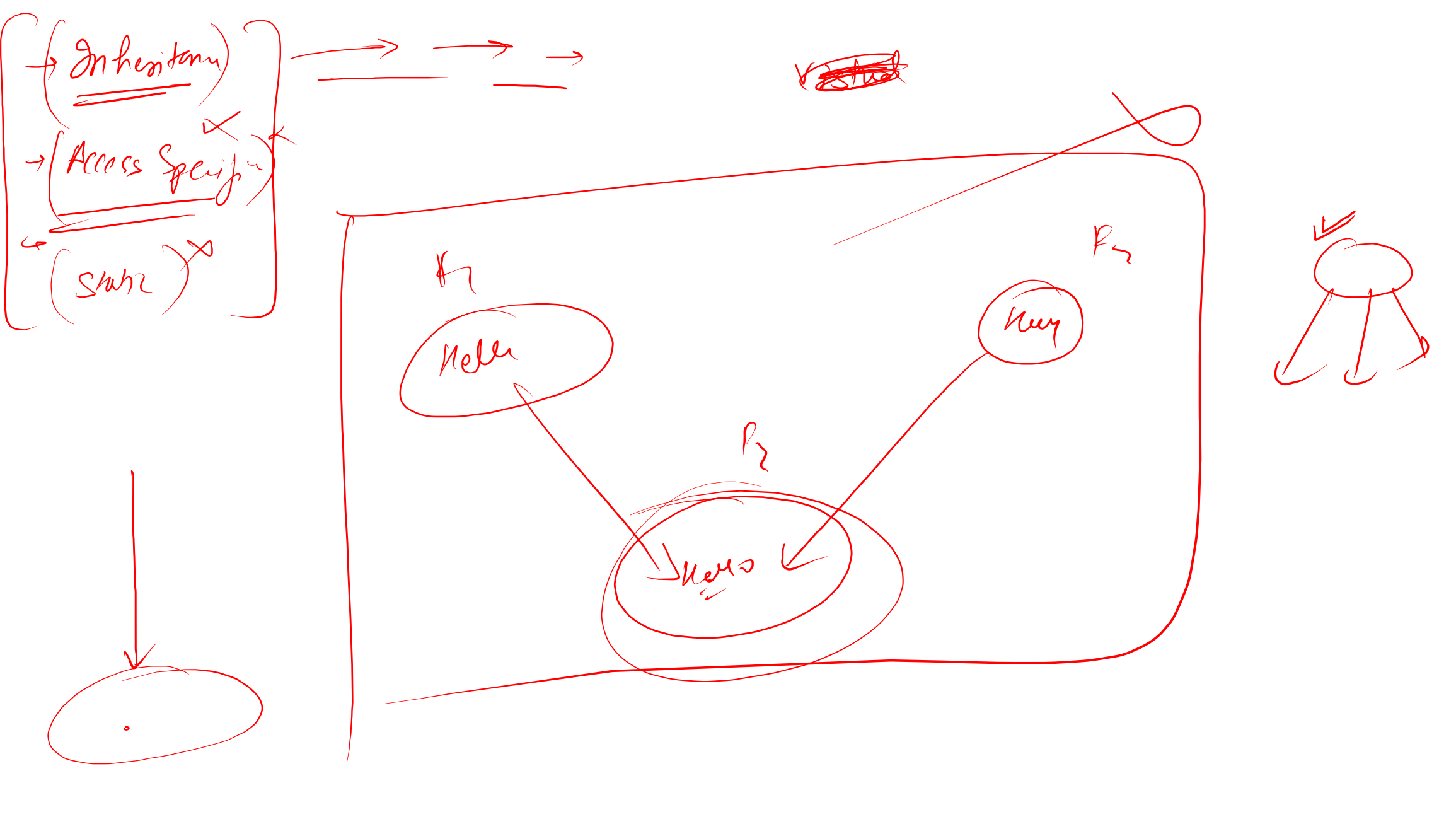


Param Const






```java
public static class Person{
    String name;
    int age;

    public void saysHi(){
        System.out.println(this.name + " ["+this.age+"] says hi");
    }

    // Person(){
    //     System.out.println("default constructor");
    //     name = "sample";
    //     age = 1;
    // }

    // Person(String n , int a){
    //     System.out.println("paramterized constructor");
    //     name = n;
    //     age = a;

    //     saysHi();
    // }

    Person(String name,int age){
        System.out.println("paramterised constructor");
        this.name = name;
        this.age = age;
    }
}
```


Inheritance
Access Specifier
Static

- → Objects & Classes
- → Memory management
- → reference / instance
- → Constructor
  - ↳ *
- → this
- → new / 3 phases
- → persist