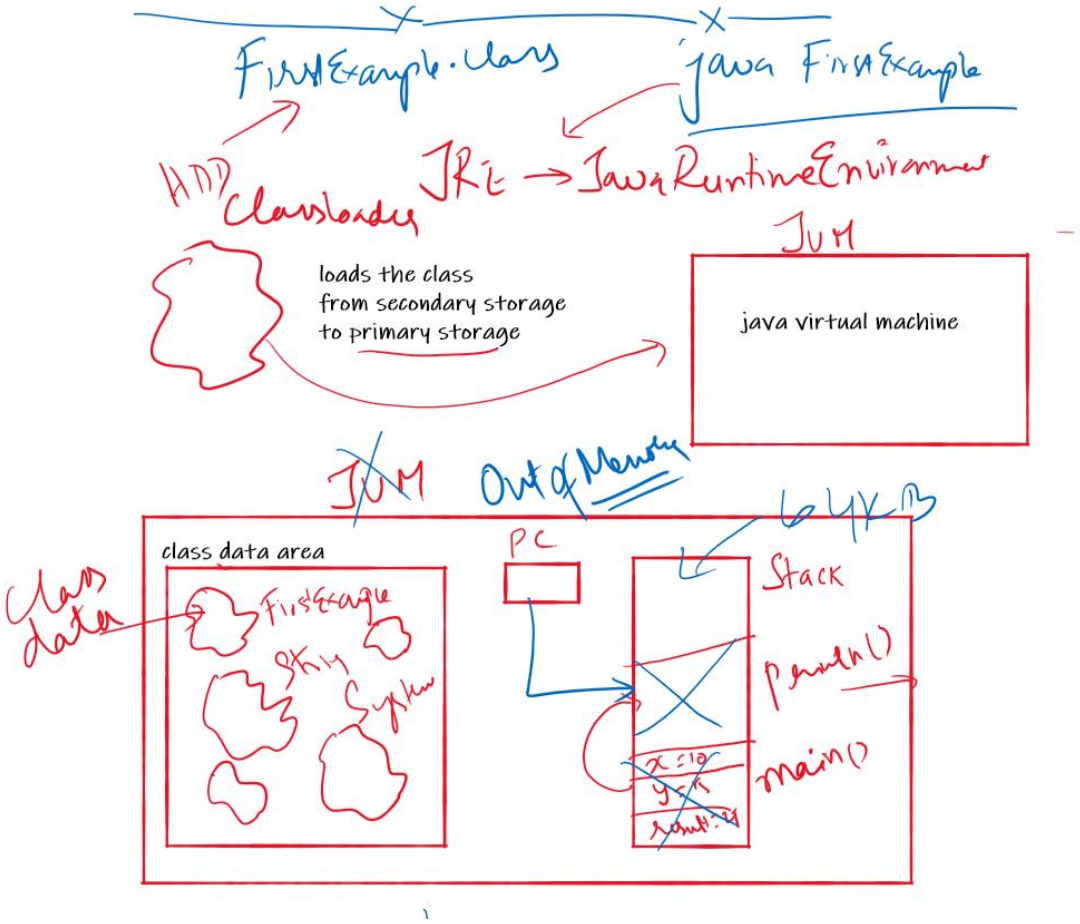
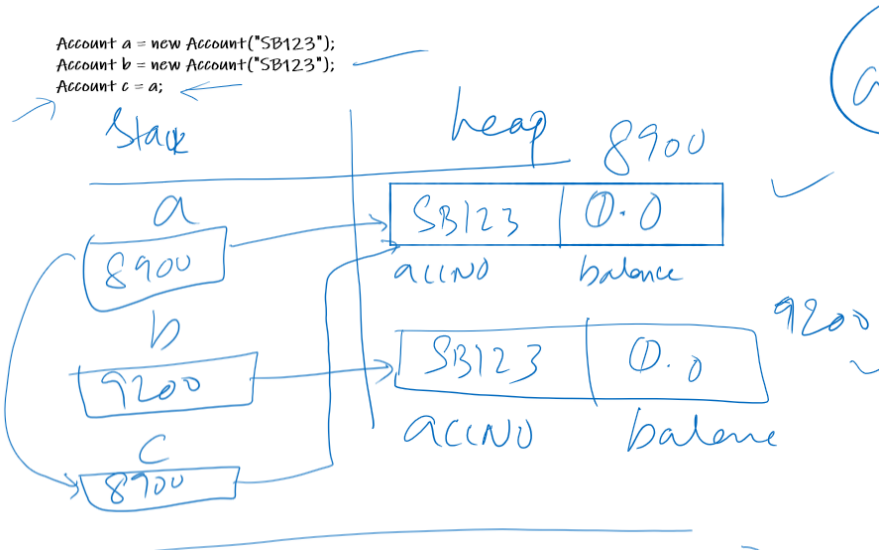
JRE and JVM:

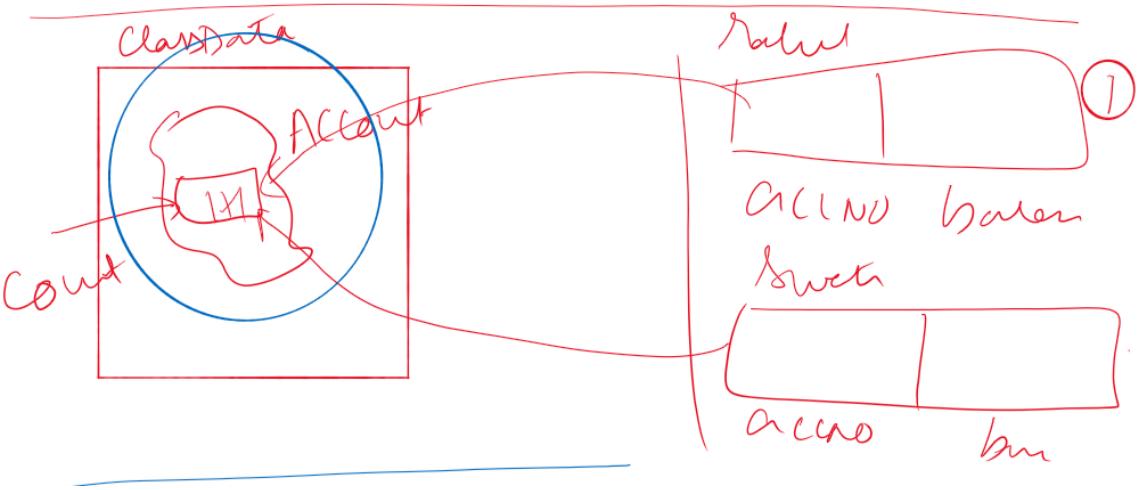


Stack and Heap:

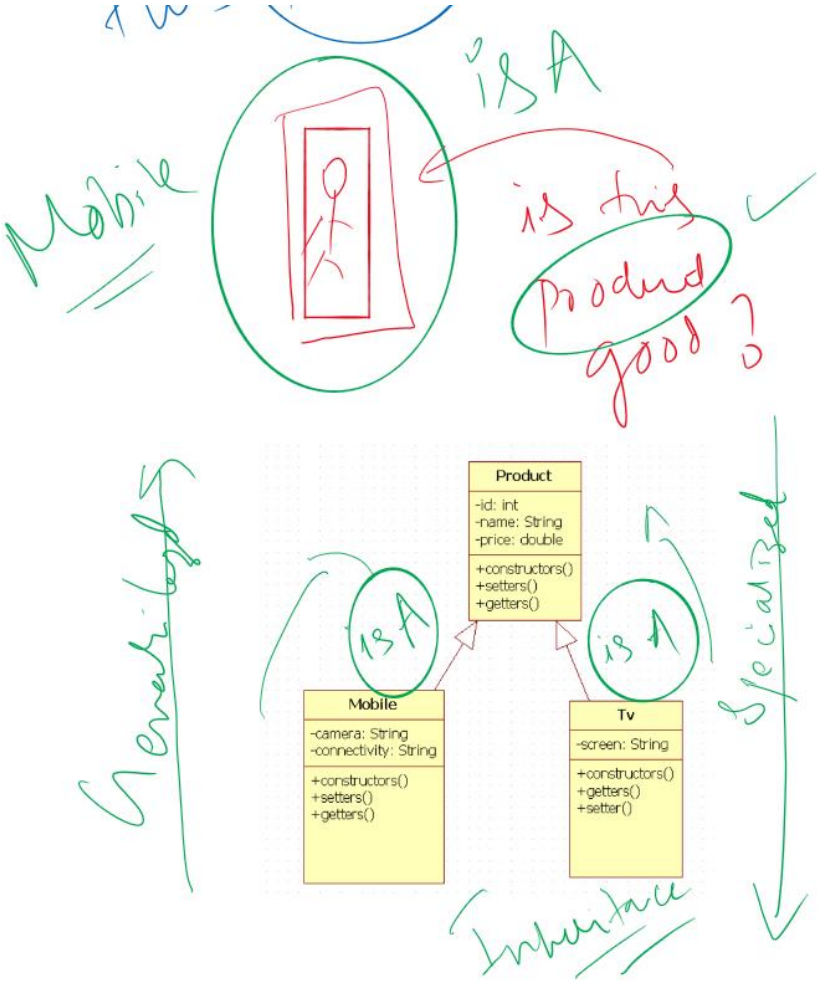


Static Variables shared by all objects of a class:

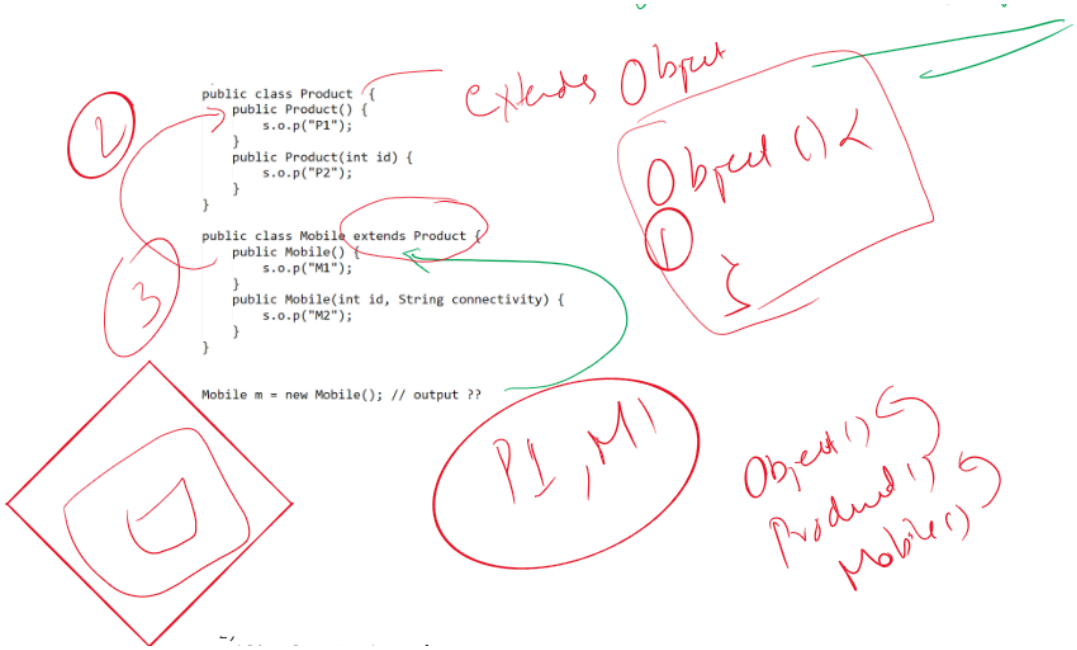
static int count ;



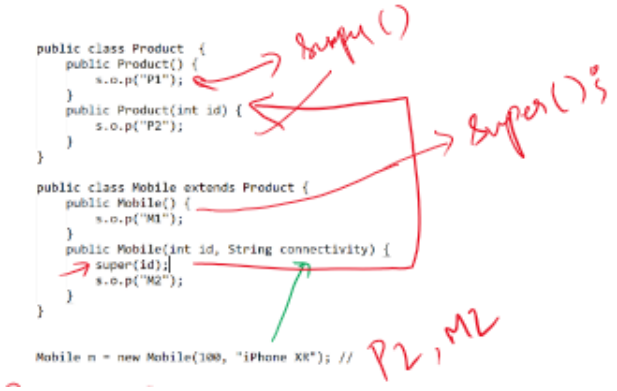
Generalization and Specialization Relationship [IS A]



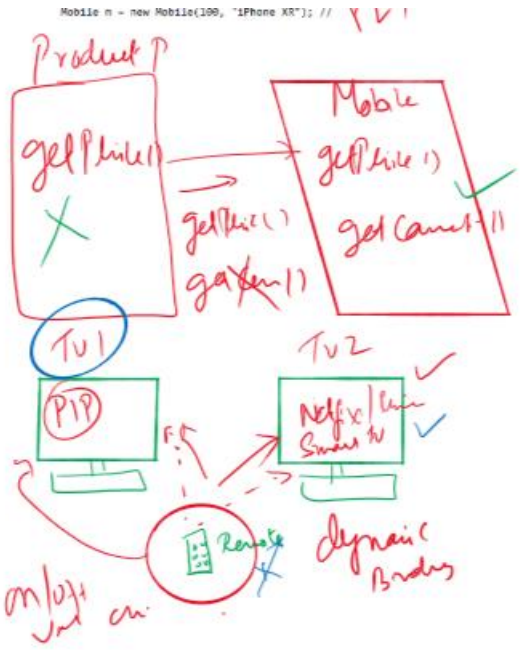
Constructor Chaining:



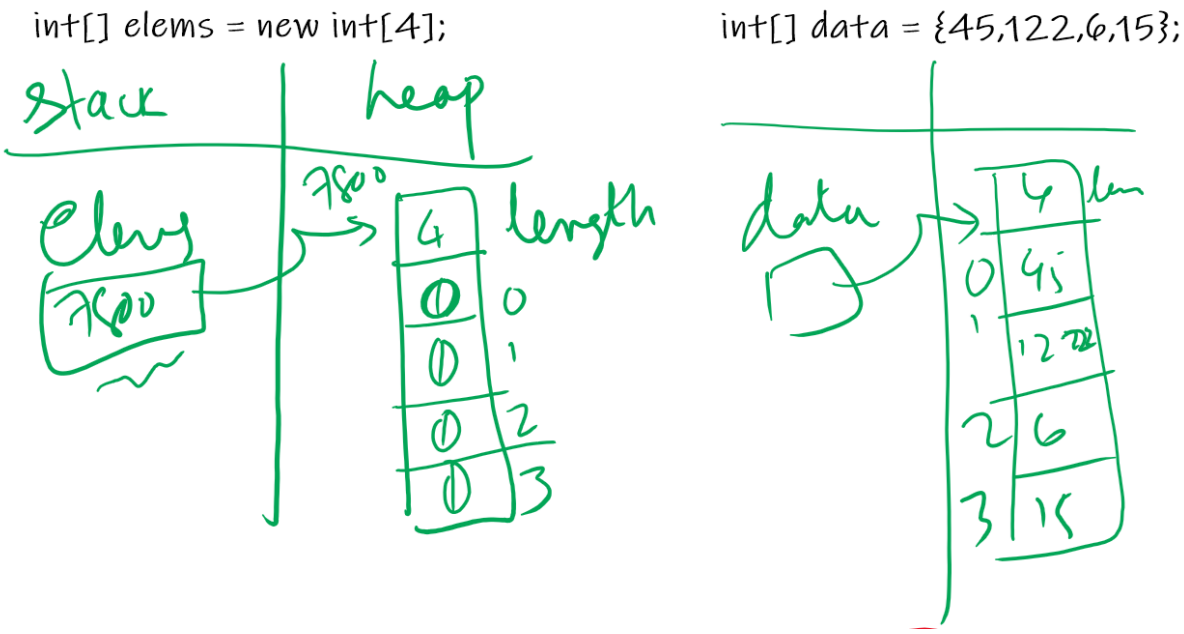

```
public class Product  {
    public Product() {
        s.o.p("P1");
    }
    public Product(int id) {
        s.o.p("P2");
    }
}

public class Mobile extends Product {
    public Mobile() {
        s.o.p("M1");
    }
    public Mobile(int id, String connectivity) {
        s.o.p("M2");
    }
}

Mobile m = new Mobile(100, "iPhone XR"); // P1, M2
```



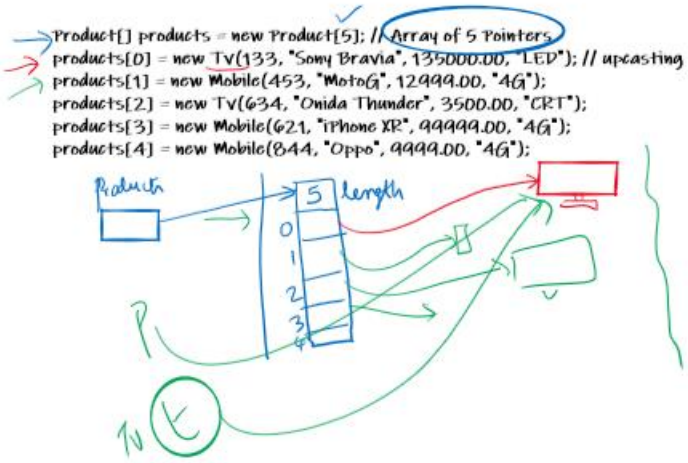
Upcasting:



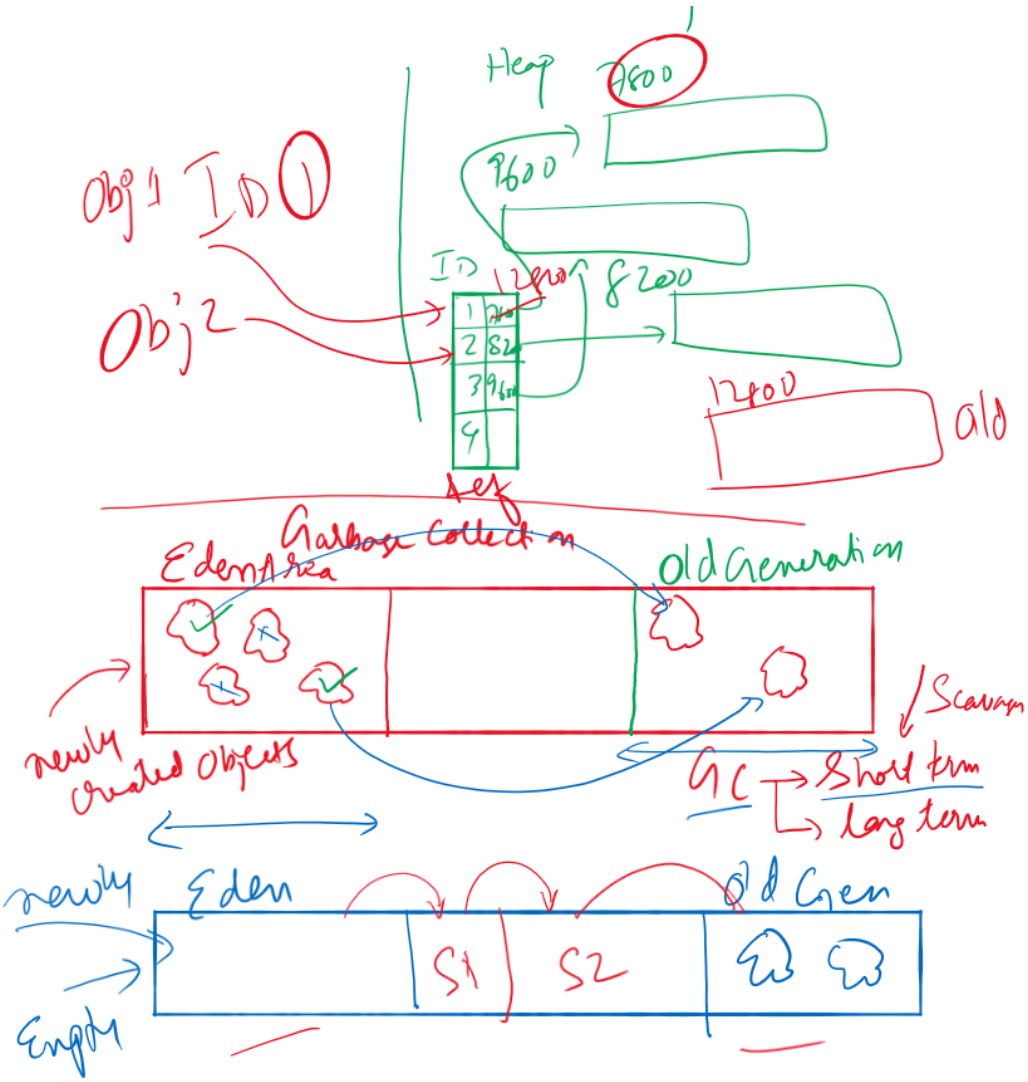
Arrays:

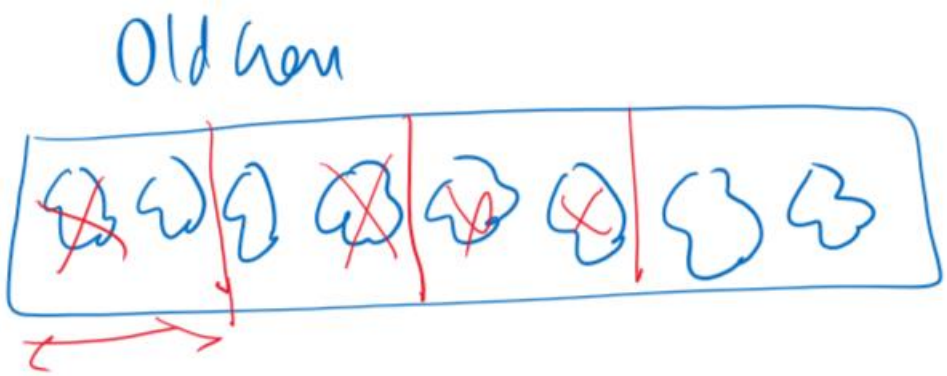


Array of Pointers:

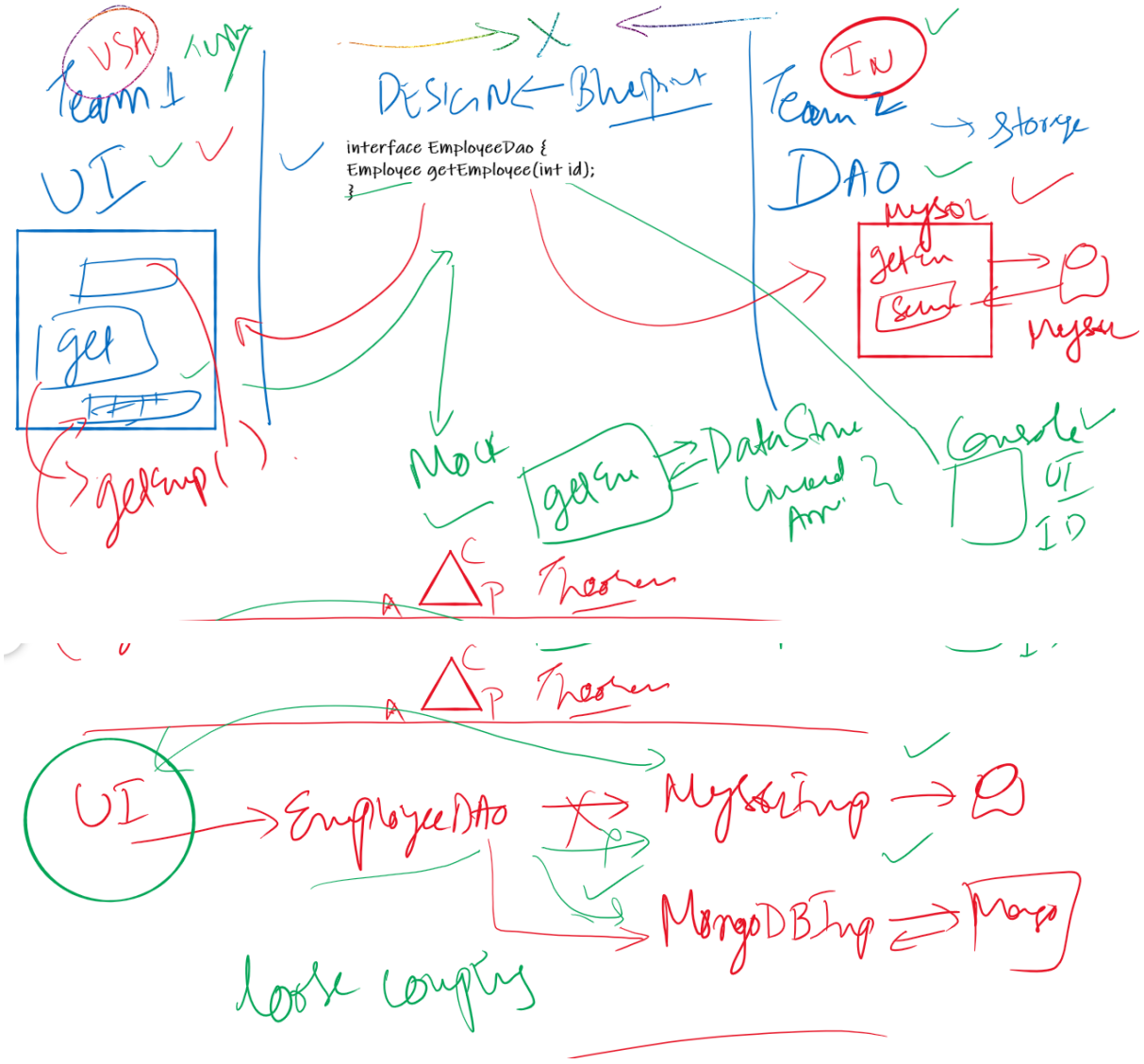


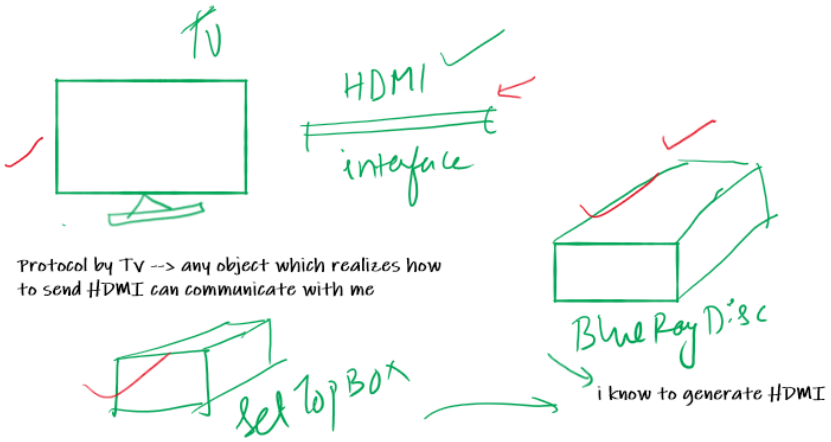
Garbage Collection and References:

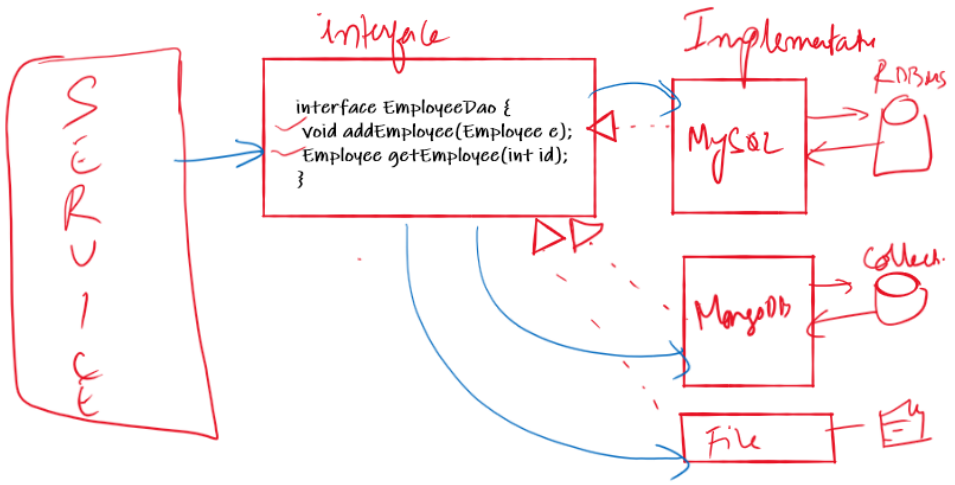




Programming to interface:




TV
HDMI
interface
Protocol by TV --> any object which realizes how to send HDMI can communicate with me
Blue Ray Disc
i know to generate HDMI
Set Top Box

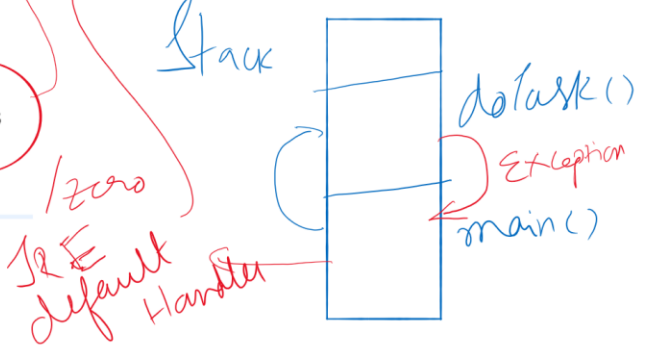

Service
interface
interface EmployeeDao {
void addEmployee(Employee e);
Employee getEmployee(int id);
}
Implementation
MySQL
RDBMS
MongoDb
Collection
File

Exception Handling:

Default Handler

```java
public class Test {

    public static void main(String[] args) {
        System.out.println("Hello !!");
            doTask();
        System.out.println("Bye!!!");
    }

    private static void doTask() {
        int x = 10;
        int y = 0;
        System.out.println("Result : " + x / y);
        System.out.println();
    }

}
```

```
Hello !!
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at Test.doTask(Test.java:13)
        at Test.main(Test.java:6)
```



Checked

1) compiler enforces programmer to handle them using try - catch syntax

2) these exceptions are a result of issues outside of JRE like [database, OS, filesystem, memory]
like "UniqueKeyConstrait", "Connection issues", "FileNotFoundException"

unchecked

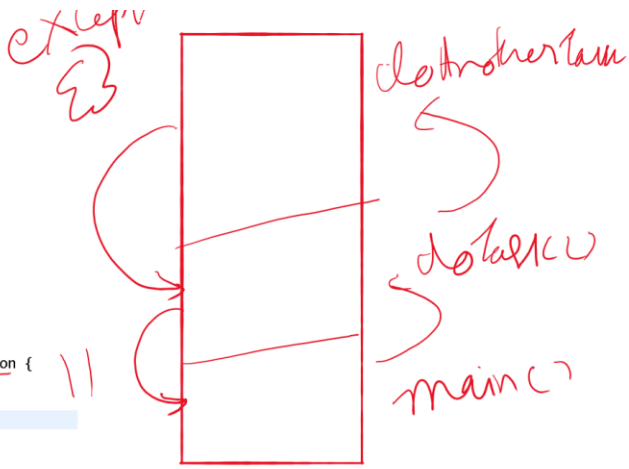
1) Compiler doesn't enforce you to handle

2) Generally these exceptions happen due to reasons within JRE

Examples:
/ 0
Index based problems
int[] elem = {5,6,3};
int data = elem[8]; //
ArrayIndexOutOfBoundsException

Product p;
p.getId(); // NullPointerException

```java
public class Test {

    public static void main(String[] args) {
        System.out.println("Hello !!");
            try {
                doTask();
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            }
        System.out.println("Bye!!!");
    }

    private static void doTask() throws FileNotFoundException {
        doAnotherTask();
    }

    private static void doAnotherTask() throws FileNotFoundException {
            FileInputStream fis = new FileInputStream("a.txt");
    }

}
```



UI	Interface	DAO Implementation class

	interface UserDao {
	register(User u);
	}
		class UserDaoMySQLImpl
		
		register(User u) {
		try {
		
		} catch(SQLException ex) {
		Print message
		}
		}

try {
 	
} catch(SQLException ex) {
Print message
}
Issue: Tight coupling: changing to MongoDB/File cacth blocks change
You are exposing to the client that SQL is used
	interface UserDao {
	register(User u) throws SQLException;
	}
		class UserDaoMySQLImpl
		
		register(User u) throws SQLException {
		
		}

try {
 	
} catch(UserExistsException ex) {
Print message
}
	interface UserDao {
	register(User u) throws UserExistsException;
	}
		class UserDaoMySQLImpl
		
		register(User u) {
		try {
		
		} catch(SQLException ex) {
		throw new UserExistsException(“users exist”);
		}
		}
		--- 	
		class UserDaoMongoImpl
		
		register(User u) {
		try {
		
		} catch(MongoException ex) {
		throw new UserExistsException(“users exist”);
		}
		}