03 October 2024 20:54

Importance of Java

1. Platform independent
2. Follows OOPS concept - Object oriented language.
3. Multithreaded
4. Robust and secure
5. Has several reliable associated frameworks for web development

OOPS Concept

1. Encapsulation :- Binding class and data together.

```
Class Car {
private String carname;
private String carmodelno;
private String carspeed;
private int carRegNo;
}
```

Cl.carname;

Class Vehicles

Car c = new Car();

use car class.

Setter methods

Getter methods.

Polymorphism

↳ something existing in multiple forms:

method overloading
method overriding.

↳ using a method with same name, we can perform different tasks.

03 October 2024 21:15

Abstraction

↳ abstract method.

Car engine → if Cook is implemented
↳ if Car multiple things

Inheritance -

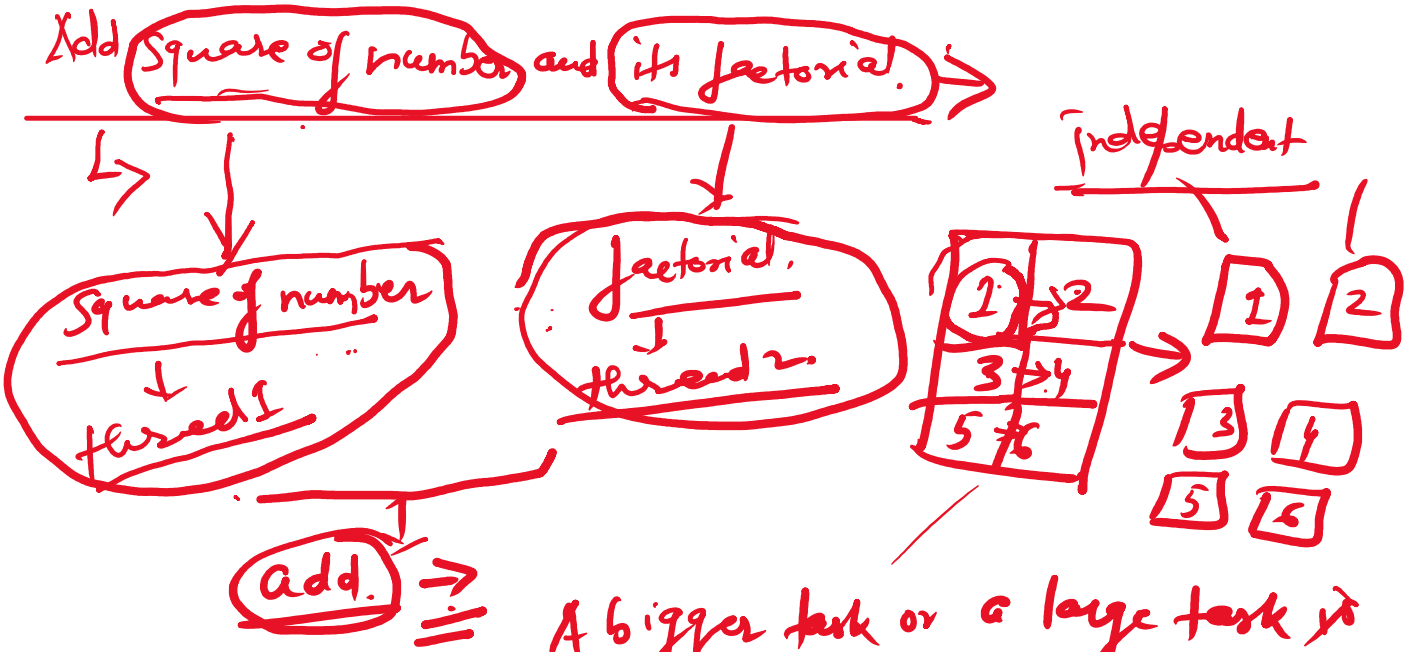
L> Parent & Child

```
class parent {
  ~~~ methods ~~~
}
class child extends parent {
}
```

Multithreaded

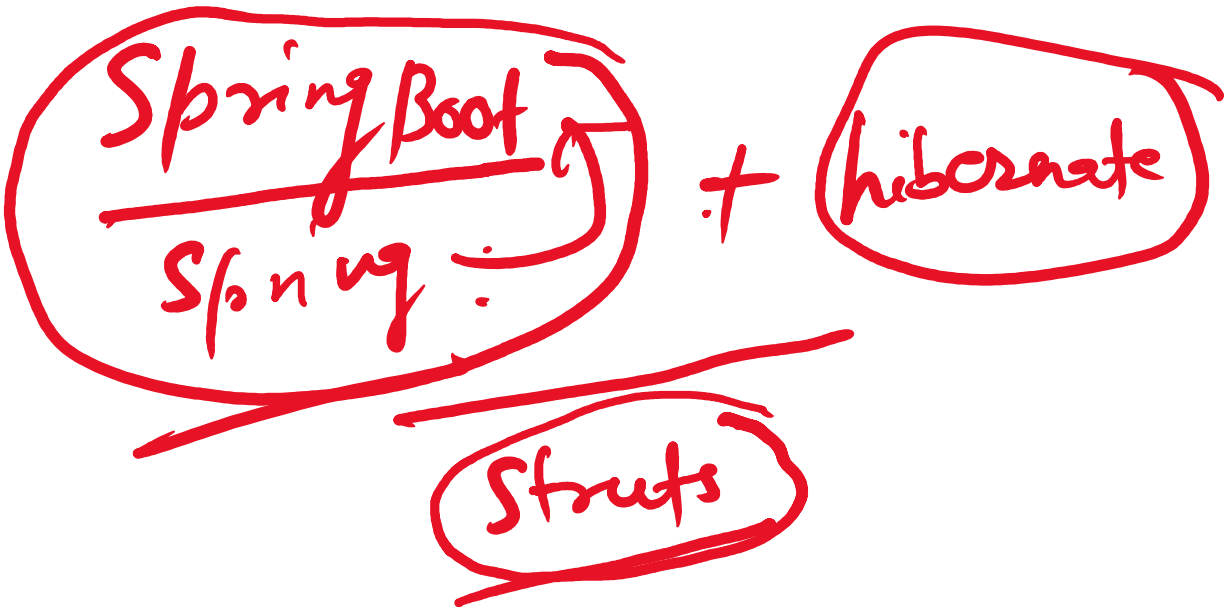
↳ Process is divided into smaller units and these units are independent.

Process → thread → find the factorial, thread → finding the squares of numbers, thread → adding factorial and Sq



A bigger task or a large task is splitted into smaller tasks and these are excuted independently.

03 October 2024 21:32



Data types and operators

Class (example)

10 + 20

Class + method

Reason

Object oriented language

Class and objects

Class is a structure or blueprint of object.

Car → Carname, Carspeed, carNumber etc.

Class

methods
variables.
constructor.

Datatypes, Operators.

- primitive data type
- Non primitive data type

→ int, char, boolean, float, double, long

- 5, 6, 7, 8, 80, 1000, 100000 (Integers)
- 'a', 'b', 'c', 'm', 'n'
- boolean → true or false.

float → 5.68, 10.87695 → 10000...

double → more than range of float → 5.68

Let x=5;

x=x²+1;

x=7 → 7²+1

=49+1=50.

Variables: → we can assign a value to a variable.

100+500 = 600

1400+1500 = 2900

↳ int a;

int b;

{ Define, declare, initialize.

int a; → define

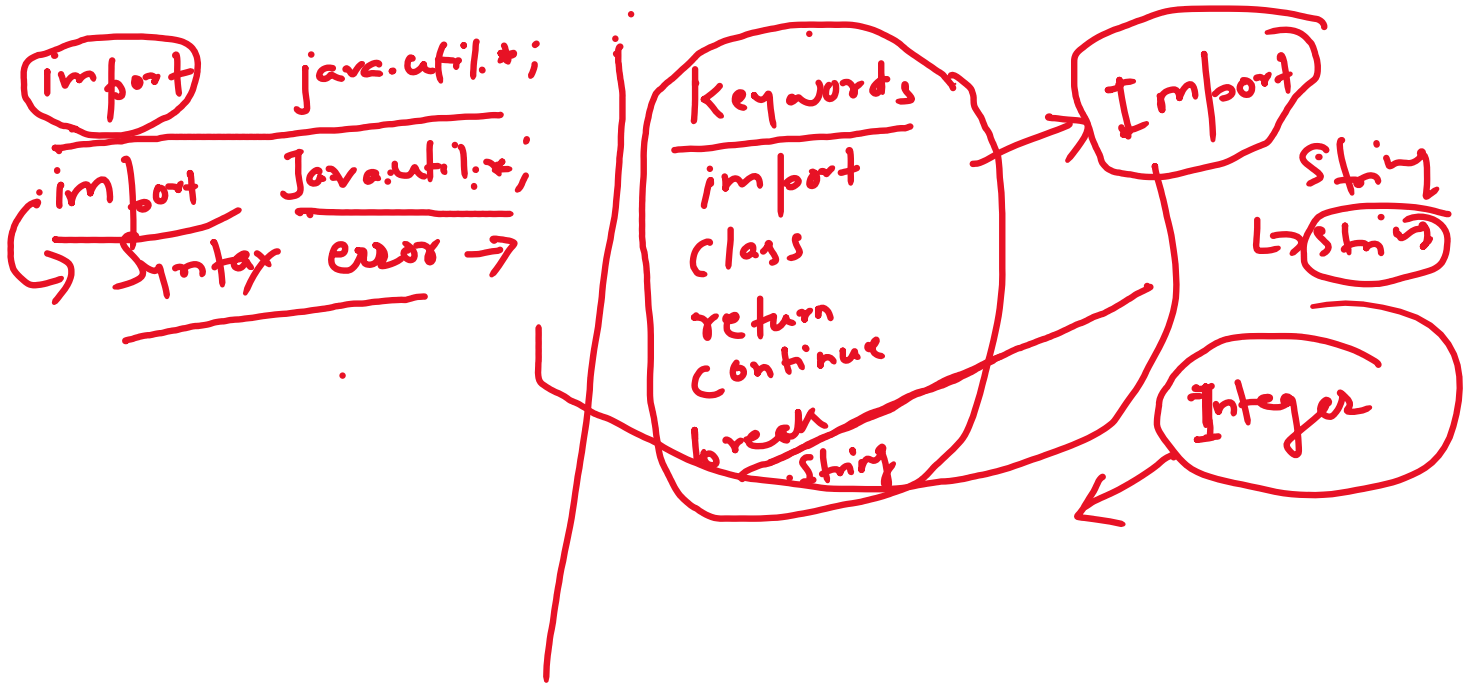
a=5; → initialization

int a=4;

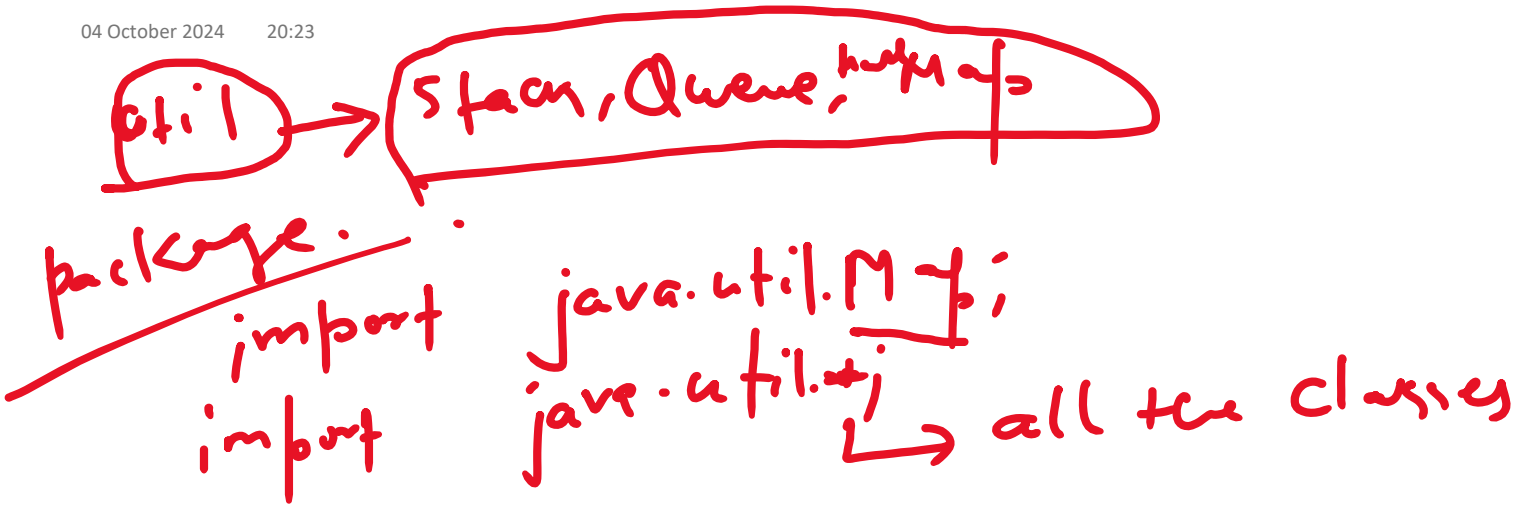
int a;

a=100;

03 October 2024 22:00



04 October 2024 20:23



04 October 2024 20:25

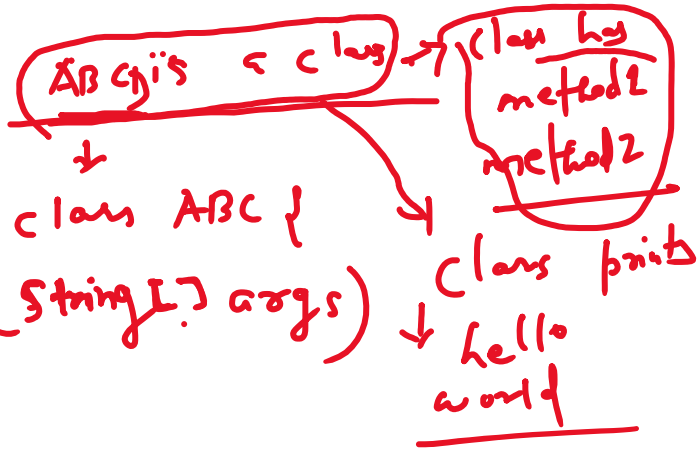
1. import java.util.*;

2. public class ABCD {

public static void main(String[] args)
{

}

}



```
public static void main (String args[]) {

System.out.println("Hello World");
System.out.print("Hello World 2");

}
}
```

04 October 2024 20:58

Output

1. Hello WorldHello World2

2. Hello World
   Hello World2

(÷)
(×)

Operators

Arithmetic Operator

↳ +, −, *, /, %

↓

+ → add
− → subtract
* → multiply

/ → quotient

% → remainder

13 / 2 → 6

13 % 2 → 1

13 + 2 = 15

13 ÷ 2

quotient → 6

remainder → 1

2) 13 (6
12
1

System.out.println (13+2);

System.out.println(50);

2. Unary Operators

++ → incremental

-- → decremental

+=

-=

```
int a=5

System.out.println(a++);
System.out.println(a);
System.out.println
    (++a);
```

output → 5
6
7

Variables

int a=5 a=7, a=8

↳ S.O.println(a) → 5, 7, 8.

Let x=5 → y = x²+1.

x: 2

y = 2²+1 = 5

Constant → S.O.println(5);

π = 3.1415

Definition:- int a; → Definition

char mychar;

a=6; → initialization

Declaration → int a=5;

int a

float a

double a

data type variable_name

↳ int
char
double
long
float