Basics of java collection api

---

readymade ds:" java collection"
aka readymade ds in java
how to use them?
performance consideration?

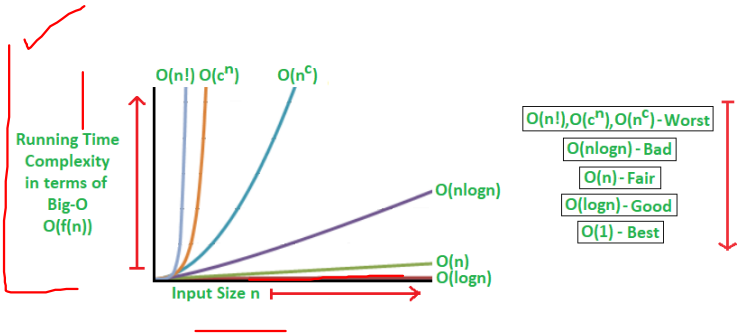
Collection
List: allow duplicate
LinkList
Vector
ArrayList
PriorityQueue

Set
HashSet: interanlly use hashing*
hashing make searching optimal operation
Big O Notation: O(1)

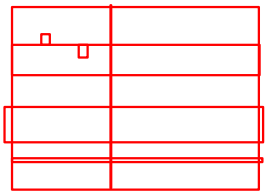
neighter sorted nor ordered

LinkedHashSet
it maintain the inerstion order
TreeSet
put the data into to sorted order

Map
HashMap
LinkedHashMap
TreeMap



Hashing:



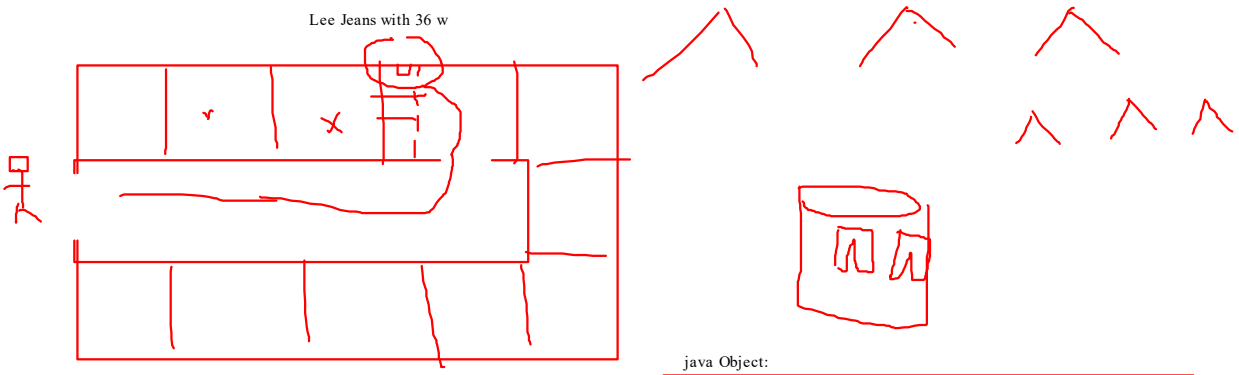
Map is a special collection
K, V
marks of student
name marks

Generics:
used with java data structure
it was added in java 5 (2004)

a
all
almost
any
any.
below
could
crow
day
a
all

Why hashing and what is that
mathmatical way to arrange data so that retival performance must be best
O(1)

Lee Jeans with 36 w



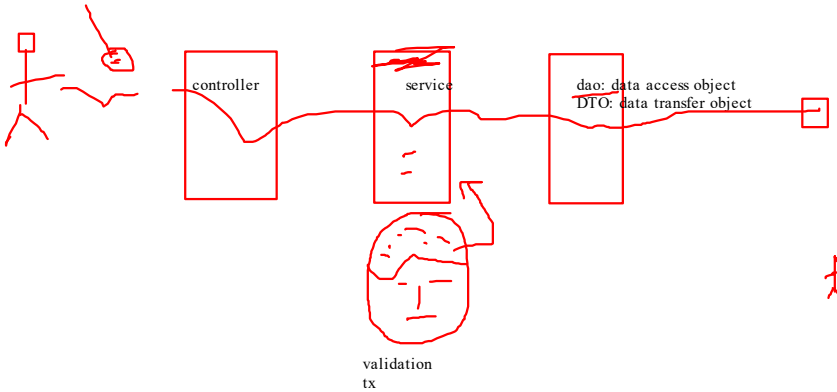
hashing is a process to divide data into the buckets for ……

java Object:

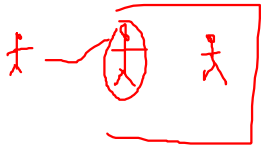
equals: if 2 objects are eq they should be from the same bucket

hashcode: if two objects have same hashcode they may be same or not

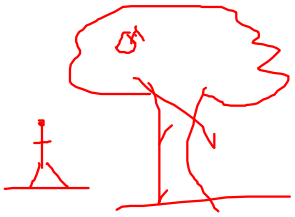
DTO



validation
tx
security
caching



```
interface Cookable{
	void cook();
}
public class C_AnnonymousInnerClass {

	public static void main(String[] args) {

		Cookable c=new Cookable(){

			@Override
			public void cook() {
				System.out.println("street food");
			}
		};
Cookable c2=new Cookable() {

			@Override
			public void cook() {
				System.out.println("street food");
			}
		};


	}

}
```



hey java cookable is interface
take it and implement it into a class
i dont care about the name of that class
just give me sigle object of that class

Java concurrency/ multithreading

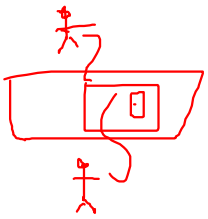
* program vs process vs threads
* how to create threads
  Thread, Running
* dead lock
* producer consumer problem
* java 5 enhancement

program vs process

software that is installed on machine
and it dont occupied RAM, CPU

process
instance of running program

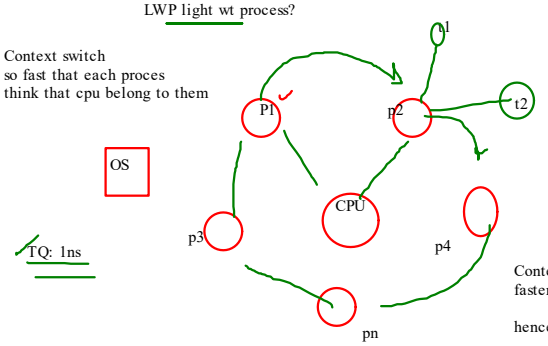
within the process u can have multiple threads



What is thread?

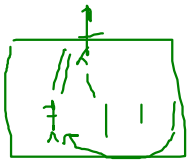
LWP light wt process?

Context switch
so fast that each proces
think that cpu belong to them



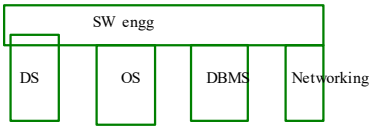
TQ: 1ns

When Context switch happen
then
os manage the information into some register



Context switch bw the threads is much
faster then Context switch bw the processes

hence it is called LWP

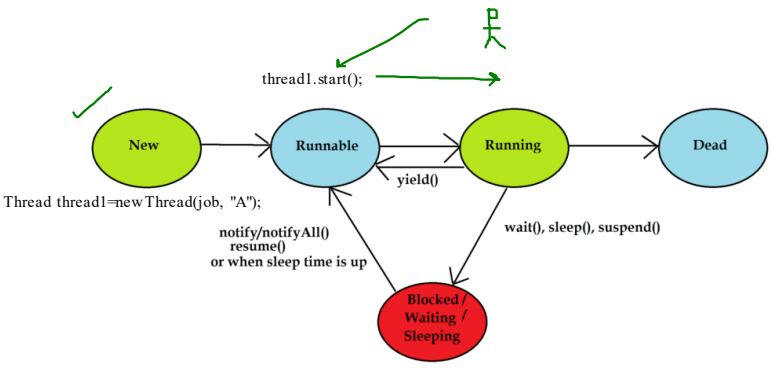


We want to create theads in java?
adv: better utilization of CPU

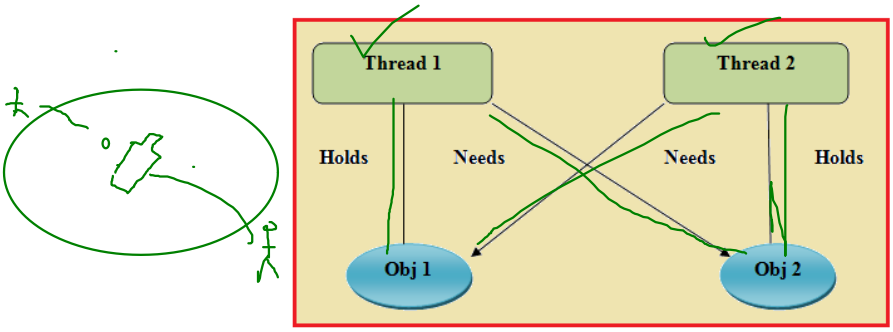
How to write multithreading code in java?

- Thread → worker
- Runnable → Job

Thread life cycle



Thread Lifecycle using Thread states

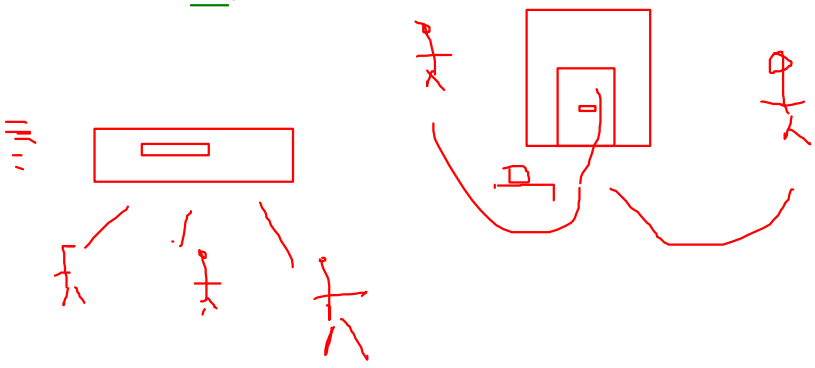


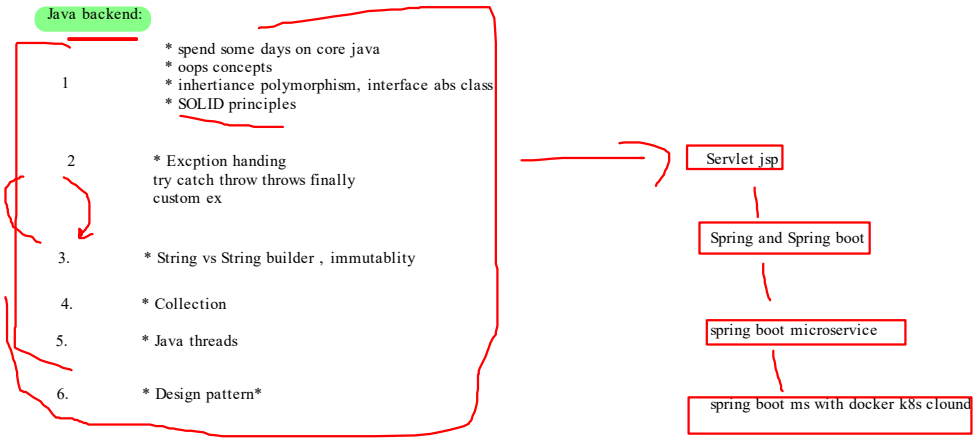

```
Object r1 = "bat";
Object r2 = "ball";
```

```
Object r1 = "bat";
Object r2 = "ball";

Thread t1 = new Thread(new Runnable() {
    @Override
    public void run() {
        synchronized (r1) {
            System.out.println("Thread t1 have taken the lock on r1");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
            }
            synchronized (r2) {
                System.out.println("Thread t1 have taken the lock on r2");
            }
        }
    }
});
```

```
Thread t2 = new Thread(new Runnable() {
    @Override
    public void run() {
        synchronized (r2) {
            System.out.println("Thread t2 have taken the lock on r2");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
            }
            synchronized (r1) {
                System.out.println("Thread t2 have taken the lock on r1");
            }
        }
    }
});
```

```
t1.start();
t2.start();
```




Java backend:
1
* spend some days on core java
* oops concepts
* inhertiance polymorphism, interface abs class
* SOLID principles
2
* Excption handing
try catch throw throws finally
custom ex
3.
* String vs String builder , immutablity
4.
* Collection
5.
* Java threads
6.
* Design pattern*
Servlet jsp
Spring and Spring boot
spring boot microservice
spring boot ms with docker k8s cloud