Java8

Thursday, 20 April 2023 12:53 AM

Java OOPS concepts
https://raygun.com/blog/oop-concepts-java/#:~:text=Abstraction%2C%20encapsulation%2C%20polymorphism%2C%20and,association%2C%20aggregation%2C%20and%20composition.

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Association - two classes are unrelated and can exist without one another.
- Aggregation - has a relationship(one-way), other can live without main object
- Composition - stricter form of aggregation, mutually dependent on each other

Streams
https://www.java67.com/2014/04/java-8-stream-examples-and-tutorial.html
https://mkyong.com/java8/java-8-stream-iterate-examples/
https://www.geeksforgeeks.org/functional-programming-in-java-8-using-the-stream-api-with-example/
https://www.javaguides.net/2018/07/java-8-stream-api.html

Examples:

```
staticclassStudent{
intid;
Stringname;
publicStudent(intid,Stringname){
this.id=id;
this.name=name;
}

Private static Map<Integer,String> convertListToMap(List<Student>
studentList){
Map<Integer,String>
map=studentList.stream().collect(Collectors.toMap(a->a.id,b->
b.name));
returnmap;
}
}


Studentst1=newStudent(1,"vg");
Studentst2=newStudent(2,"venk");
List<Student>studentList=newArrayList<>();
studentList.add(st1);
studentList.add(st2);

Student.convertListToMap(studentList).entrySet().stream().forEach(a->
System.out.println("a:"+a.getKey()+","+a.getValue()));

Stringval=Stream.iterate(new int[]{0,1},n->new int[]{n[1],n[0]+n[1]})
.limit(10)
.map(n->String.valueOf(n[0]))
.collect(Collectors.joining(","));
System.out.println("fibonacci:"+val);

List<Integer>list=Stream.iterate(0,n->n+1)
.limit(10)
.filter(a->a%2==1)
.collect(Collectors.toList());
list.forEach(a->System.out.print(a+","));
```