

Part A- OOP Class Design Guidelines

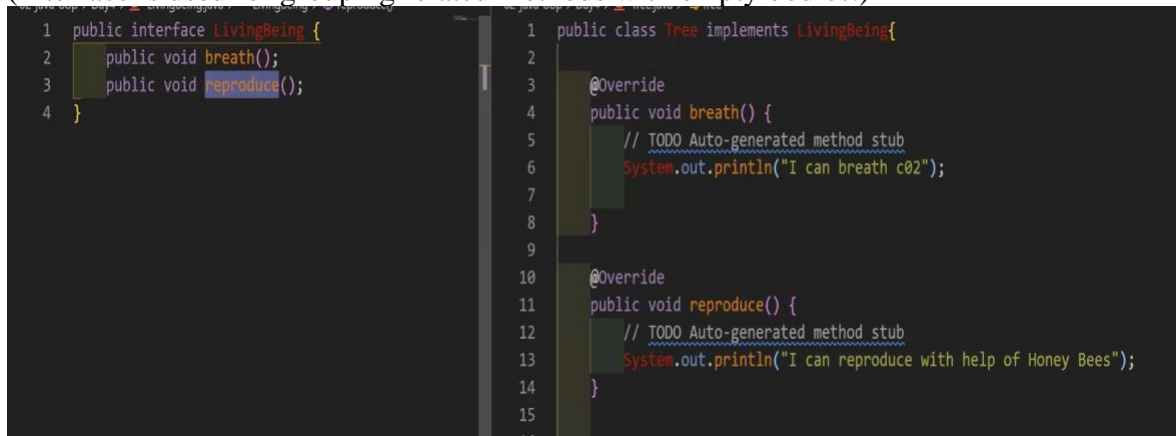
Question: Please choose 3 guidelines and discuss them in-depth.

OOP- object-oriented programming is a paradigm which is a way of programming that organize data and modifiers to design products or applications.

Encapsulation – Encapsulation is to use for hiding complexity not details. (you want to hide complexity from any user). Grouping functionality into a class that you will reuse and complicated processing can be encapsulated into a class and simple methods exposed to your other class For example, by implementing getters and setters, attributes are accessible from any other class.

Inheritance- By using “extends” keyword in child class, child class can access everything from parent class. In other words, inheritance is saving programmers’ time by using “DRY” don’t repeat yourself method.

Interface- Interface cannot contain constructor and interface methods do not have a body. (Interface is used for grouping related methods with empty bodies.)



```
1 public interface LivingBeing {
2     public void breath();
3     public void reproduce();
4 }
5
6 public class Tree implements LivingBeing{
7     @Override
8     public void breath() {
9         // TODO Auto-generated method stub
10        System.out.println("I can breath c02");
11    }
12
13    @Override
14    public void reproduce() {
15        // TODO Auto-generated method stub
16        System.out.println("I can reproduce with help of Honey Bees");
17    }
18 }
```

According to the above screenshot, Tree class can access methods and overriding those from the LivingBeing class with “implements” keyword in Tree class.

Polymorphism – method overloading and constructor overloading. Can be used by other class, not only for one class.