

Question 1:

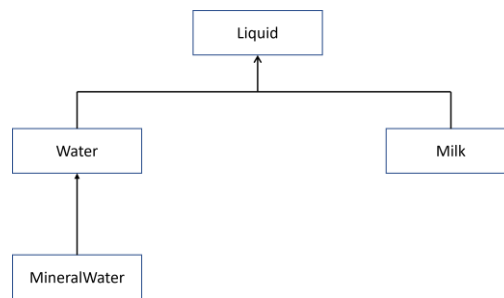


Figure 1

```
1 import java.util.*;
2
3 class Liquid {
4     String name;
5     double price;
6
7     Liquid(String name, double price) {
8         this.name = name;
9         this.price = price;
10    }
11
12    public String toString() {
13        return name + " " + price;
14    }
15 }
16
17 class Water extends Liquid {
18     public Water(String name, double price) {
19         super(name, price);
20     }
21 }
22
23 class MineralWater extends Water {
24     MineralWater(String name, double price) {
25         super(name, price);
26     }
27 }
28
29 class Milk extends Liquid {
30     Milk(String name, double price) {
31         super(name, price);
32     }
33 }
34
35 public class Wildcard {
36
37     public static void deleteLiquid(List<? extends Water> list, Water water) {
38         list.remove(water);
39     }
40
41     public static void addLiquid(List<? super MineralWater> list) {
42         list.add(new MineralWater("Mineral Water", 10));
43     }
44
45     public static void print(List<?> list) {
46         for (Object item : list)
47             System.out.println(item);
48     }
49
50     public static void main(String[] args) {
51
52         List<Liquid> LiquidList= new ArrayList<Liquid>();
53         List<MineralWater> MineralWaterList= new ArrayList<MineralWater>();
54
55         addLiquid(LiquidList);
56
57         addLiquid(MineralWaterList);
58         addLiquid(MineralWaterList);
59
60         Water Liquid = MineralWaterList.get(0);
61
62         deleteLiquid(MineralWaterList, Liquid);
63
64         print(MineralWaterList);
65     }
66 }
```

Figure 2

Question 2:

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 @FunctionalInterface
5 interface Utility{
6     public abstract void printList(ArrayList<Player> p);
7 }
8
9 enum PlayerCategory{
10     GOALKEEPER(0),
11     MIDFIELDER(5),
12     STRIKER(10),
13     DEFENDER(2);
14
15     int jerseyNumber;
16
17     public PlayerCategory(int jerseyNumber){
18         this.jerseyNumber = jerseyNumber;
19     }
20 }
21
22 public class Player{
23
24     public String name;
25     public PlayerCategory category;
26     public int goalsScored;
27
28     public Player(String name, PlayerCategory category, int goalsScored){
29         this.name = name;
30         this.category = category;
31         this.goalsScored = goalsScored;
32     }
33
34     @Override
35     public String toString(){
36         return name + " " + category + " " + category.jerseyNumber + " " + goalsScored;
37     }
38
39     public static void main(String [] args){
40
41         ArrayList<Player> players = new ArrayList<Player>();
42         Scanner scan = new Scanner();
43
44         while(true){
45
46             System.out.print("Enter player name: ");
47             String name = scan.next();
48
49             System.out.print("1.GOALKEEPER\n2.MIDFIELDER\n3.STRIKER\n4.DEFENDER\nChoose player position: ");
50             int categoryValue = scan.nextInt();
51             PlayerCategory playerCategory;
52
53             for(PlayerCategory category : PlayerCategory.values()) {
54                 if(category.ordinal() + 1 == categoryValue){
55                     playerCategory = category;
56                     break;
57                 }
58             }
59             if(playerCategory == null) return;
60
61             System.out.print("Enter goals scored by " + name+": ");
62             int goalsScored = scan.nextInt();
63
64             players.add(Player(name, playerCategory, goalsScored));
65
66             System.out.print("Want to add more players? Yes/No: ");
67             if(scan.next() == "No") break;
68         }
69
70         players.sort((p1, p2) -> p1.goalsScored > p2.goalsScored ? 0 : 1);
71
72         Utility obj = p3 -> {for(Player p : p3) System.out.println(p); };
73
74         obj.printList(players);
75     }
76 }
```

Figure 3

```
$ java Player
Enter player name: Messi
1.GOALKEEPER
2.MIDFIELDER
3.STRIKER
4.DEFENDER
Choose player position: 3
Enter goals scored by Messi: 10
Want to add more players? Yes/No: Yes
Enter player name: Buffon
1.GOALKEEPER
2.MIDFIELDER
3.STRIKER
4.DEFENDER
Choose player position: 1
Enter goals scored by Buffon: 2
Want to add more players? Yes/No: Yes
Enter player name: Xavi
1.GOALKEEPER
2.MIDFIELDER
3.STRIKER
4.DEFENDER
Choose player position: 2
Enter goals scored by Xavi: 6
Want to add more players? Yes/No: Yes
Enter player name: Maldini
1.GOALKEEPER
2.MIDFIELDER
3.STRIKER
4.DEFENDER
Choose player position: 4
Enter goals scored by Maldini: 3
Want to add more players? Yes/No: No
Messi STRIKER 10 10
Buffon GOALKEEPER 0 2
Xavi MIDFIELDER 5 6
Maldini DEFENDER 2 3
```

Figure 4

Question 3:

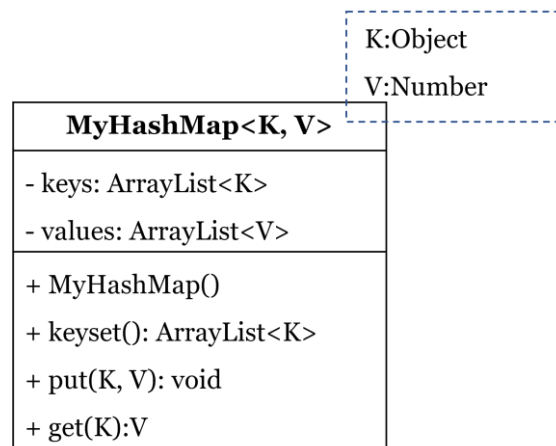


Figure 5

Question 4:

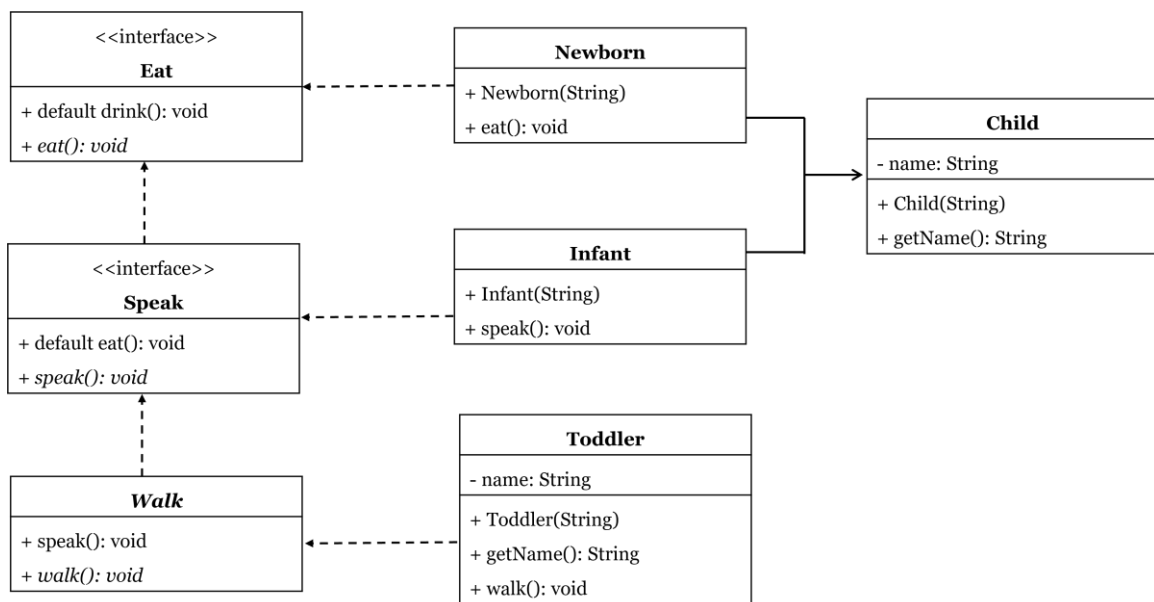


Figure 6

Question 5:

Employee
+ name: String + designation: String + salary: int + manages: LinkedList<Employee>
+ Employee(String, String, int) + addSubordinate(): void + static findSubordinates(Employee, Employee): boolean

Figure 7

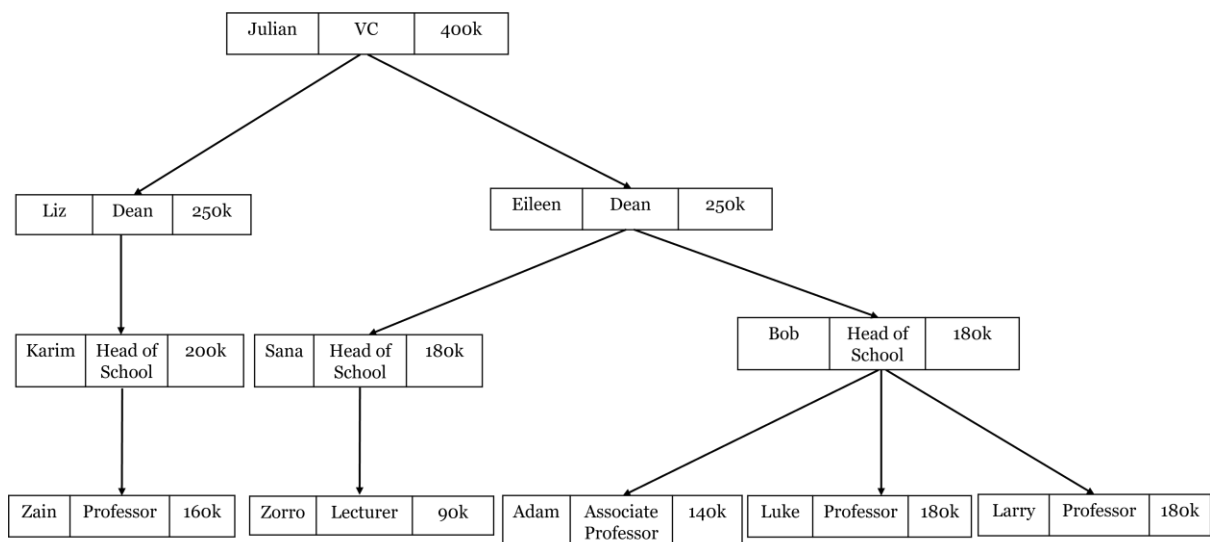


Figure 8