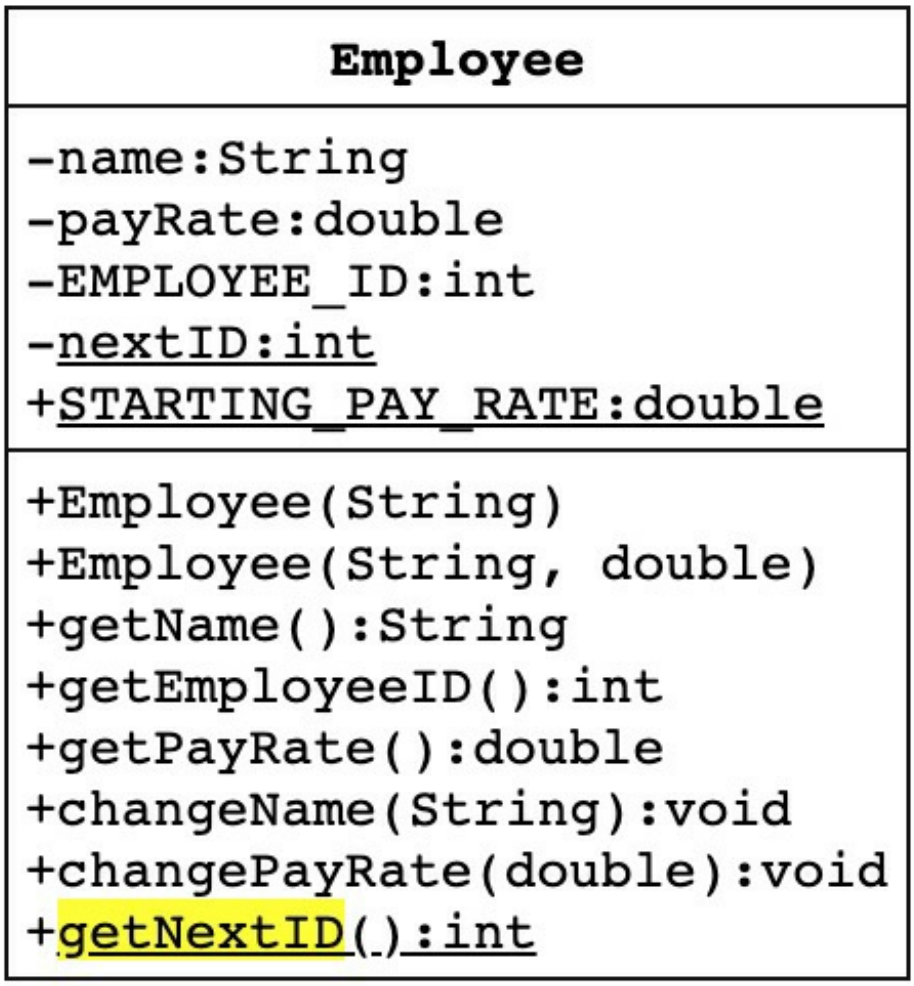
Design

1. Employee Class

where:
1)**visibility** (**+** for **public**, **-** for **private**)
2)**attribute** = data member (aka field)
3)**operation** = method (or constructor)

Note:
1)The **arg** list is a list of parameter types (e.g., int, double, String); parameter names are not included in the UML class diagram
2)Methods that don't return a value (i.e. void methods) should give a return type of void
3)Class (i.e. static) methods and fields are indicated by **underlining**
4)Constant (i.e. final) fields are indicated via naming convention: constants should be in ALL_CAPS

UML diagram



Factory Pattern

```
1.  public interface Shape {
2.      void draw();
3.  }
4.  public class CircleShape implements Shape {
5.      public CircleShape() {
6.          System.out.println(  "CircleShape: created");
7.      }
8.
9.      @Override
10.     public void draw() {
11.         System.out.println(  "draw: CircleShape");
12.     }
13. }
14.
15. public class RectShape implements Shape {
```

```
16.      public RectShape() {
17.         System.out.println(  "RectShape: created");
18.      }
19.      @Override
20.      public void draw() {
21.         System.out.println(  "draw: RectShape");
22.      }
23.  }
24.
25.  public class TriangleShape implements Shape {
26.      public TriangleShape() {
27.          System.out.println(  "TriangleShape: created");
28.      }
29.      @Override
30.      public void draw() {
31.          System.out.println(  "draw: TriangleShape");
32.      }
33.
34.  }
35.
36.   public class ShapeFactory {
37.            public static final String TAG = "ShapeFactory";
38.            public static Shape getShape(String type) {
39.                Shape shape = null;
40.                if (type.equalsIgnoreCase("circle")) {
41.                    shape = new CircleShape();
42.                } else if (type.equalsIgnoreCase("rect")) {
43.                    shape = new RectShape();
44.                } else if (type.equalsIgnoreCase("triangle")) {
45.                    shape = new TriangleShape();
46.                }
47.                return shape;
48.            }
49.  }
50.
```

Java Code

```
1.  public class Employee {
2.
3.      private String name;
4.      private double payRate;
5.      private final int EMPLOYEE_ID;
6.
```

```
    private static int nextID = 1000;

    public static final double STARTING_PAY_RATE = 7.75;

    public Employee(String name) {
        this.name = name;
        EMPLOYEE_ID = getNextID();
        payRate = STARTING_PAY_RATE;
    }

    public Employee(String name, double startingPay) {
        this.name = name;
        EMPLOYEE_ID = getNextID();
        payRate = startingPay;
    }

    public String getName() { return name; }
    public int getEmployeeID() { return EMPLOYEE_ID; }
    public double getPayRate() { return payRate; }
    public void changeName(String newName) { name = newName; }
    public void changePayRate(double newRate) { payRate = newRate; }

    public static int getNextID() {
        int id = nextID;
        nextID++;
        return id;
    }
}
```