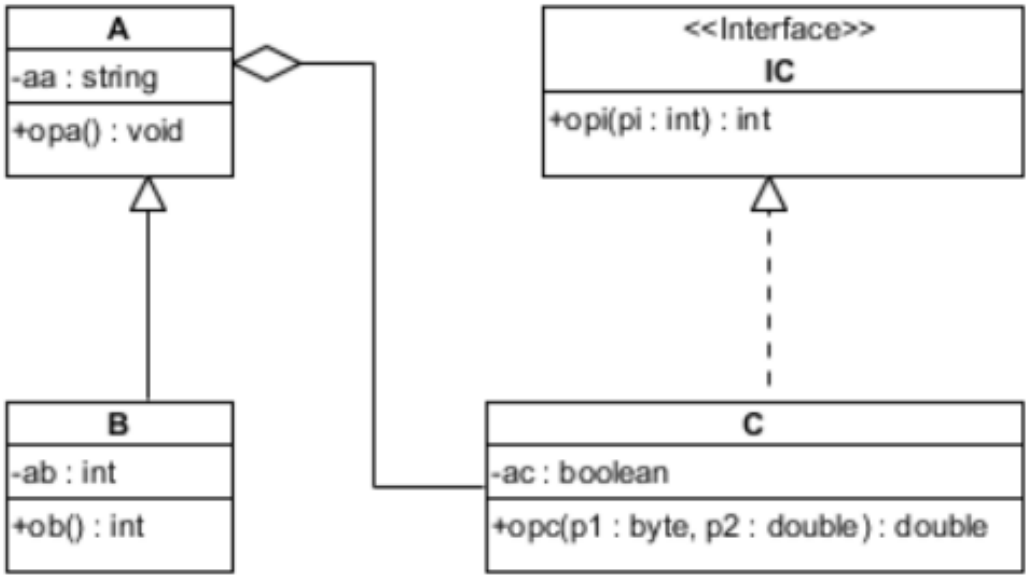
PROBLEM: Consider the following class diagram:



1. Describe the classes (name, attributes, operations).

2. Describe class relations (relation type, participating classes and their roles).

3. Write the Java code that results from the diagram (excepting the constructor).

SOLUTION

1. Describe the classes (name, attributes, operations).

Interface IC

Operations:

- opi(), visibility public, parameter pi of type int, return type int

Class A

Attributes:

- aa, visibility private, of type string

- collection with objects of type C

Operations:

- opa(), visibility public, no parameters, return type void

Class B

Attributes:

- aa inherited

- collection with objects of type C, inherited

- ab, visibility private, of type int

Operations:

- opa() inherited

- ob(), visibility public, no parameters, return type int

Class C

Attributes:

- ac, visibility private, of type boolean

- object of type A

Operations:

- implements opi()

- opc(), visibility public, parameter p1 of type byte, parameter p2 of type double, return type double

2. Describe class relations (relation type, participating classes and their roles).

Generalization between classes A and B, A is superclass and B is subclass.

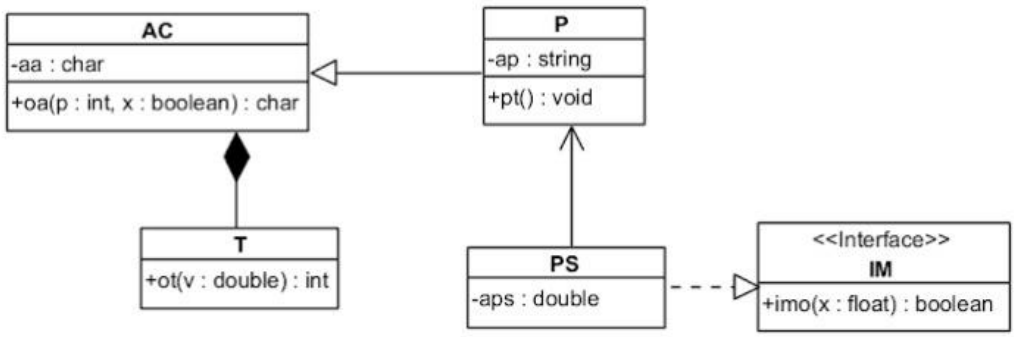
Aggregation between classes A and C, A is the aggregate and C is the component.

Realization between interface IC and class C, C implements IC.

3. Write the Java code that results from the diagram (excepting the constructor).

```
public class A {
private string aa;
private Collection<C> cc;
public void opa(){...}
}
public class B extends A {
private int ab;
public int ob(){...}
}
public interface IC {
public int opi(int pi);
}
public class C implements IC {
private boolean ac;
private A anA;
public int opi(int pi){...}
public double opc(byte p1, double p2){...}
}
```

Consider the following class diagram:



1. Describe the classes (name, attributes, operations).

2. Describe class relations (relation type, participating classes and their roles).

3. Write the Java code that results from the diagram (excepting the constructor).

1. Describe the classes (name, attributes, operations).

Interface IM
Operations:
-imo(), visibility public, parameter x of type float, return type boolean

Class AC

Attributes:
- aa, visibility private, type char

- it could have either a collection or an entity of cass T
Operations:

-oa(), visibulity public, parameter p of type int and x a boolean, return type char

Class T
Attributes:

-object of type AC

Operations:

-implements oa()

-ot(), visibility public, parameter v of type double, return type int

Class P

Attributes:
-aa inherited

-collection of objects of type T, inherited

-ap, visibility private, type string

Operations:

-implements oa()

-oa(), visibility public, no parameters, return type void

Class PS

Attributes:

-aps, visibility private, type double

-reference to class P

Operations:

-implements imo()

2. Describe class relations (relation type, participating classes and their roles).

Generalization between class AC and class P, AC is the superclass and P is the subclass.
Realization between PS and IM, PS implements IM.

Composition between class AC and T, AC is the aggregate and T is the component.
Unidirectional association beteween PS and P, PS is owning side of the relation.

3.

```
// Interface IM
interface IM {
    public boolean imo(float x);
}
// Class AC
class AC {
    private char aa;
    private T t;
    public char oa(int p, boolean x) {
        // Implementation
        return 'a'; // Placeholder return value
    }
}
// Class T
class T {
    private AC ac;
    public void oa() {
        // Implementation
```

```
 }
    public int ot(double v) {
        // Implementation
        return 0; // Placeholder return value
    }
}
// Class P
class P extends AC implements IM {
    private String ap;
    public void oa() {
        // Implementation
    }
    public boolean imo(float x) {
        // Implementation
        return false; // Placeholder return value
    }
}
// Class PS
class PS implements IM {
    private double aps;
    private P pRef;
    public boolean imo(float x) {
        // Implementation
        return false; // Placeholder return value
    }
}
```