**Paradigmas de Programação I – 3º. semestre**

**Prof. Dr. Daniel Facciolo Pires**

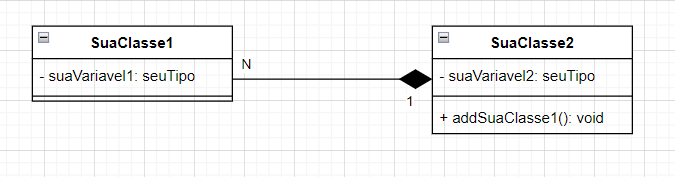
**Valor: 5,0 pts**

**Data: 15/04/2024**

**1ª. Questão – 2,0 pts**

Implemente o Diagrama de Classe a seguir. No local das palavras SuaClasse1 e SuaClasse2, crie nomes de classe que façam sentido para o tipo de associação (não vale criar classes com o mesmo nome feito em aula). Ainda, nos locais das suaVariavel1 e suaVariavel2, crie variáveis que façam sentido para a classe (não vale criar variáveis com o mesmo nome feito em aula).

Crie as variáveis necessárias, inclusive para o tipo de associação ocorrer adequadamente. Crie somente os construtores das classes. Não é necessário fazer os getters, setters e toString(). Crie o método addSuaClasse1(seuTipo suaVariavel1) na classe SuaClasse2. Crie uma classe Testa para instanciar a classe SuaClasse2, e dar um exemplo de chamada do método addSuaClasse1



**2ª. Questão – 2,0 pts**

Implemente o Diagrama de Classe a seguir. No local das palavras SuaSuperClasse, SuaSubClasse1 e SuaSubClasse2, crie nomes de classe que façam sentido para o tipo de associação (não vale criar classes com o mesmo nome feito em aula). Ainda, nos locais das suaVariavel1, suaVariavel2 e suaVariavel3, crie variáveis que façam sentido para a classe (não vale criar variáveis com o mesmo nome feito em aula).

Crie os construtores, não é preciso implementar os getters e setters. Implemente os métodos toString(). Finalmente, cria uma classe Testa e exemplifique o uso de um objeto da classe SuaSuperClasse chamando o método toString() das classes filhas (polimorfismo).

Diagrama

Descrição gerada automaticamente

**3ª. Questão – 0,5 pto**

Analise as afirmações abaixo a respeito de herança:

I. Um objeto de uma superclasse pode receber um objeto de uma subclasse

(objSuperClasse = objSubClasse)

II. Um exemplo de polimorfismo generalizado é apresentado a seguir:

Superclasse obj;

obj = new SubClasse1()

obj.toString()

obj = new SubClasse2()

obj.toString()

III. Para que possamos explorar o polimorfismo na herança, é necessário de anulação de métodos

Estão corretas:

a. Apenas III

b. Apenas I e II

c. Apenas II e III

d. I, II e III

**4ª. Questão – 0,5 pto**

Analise as afirmações abaixo sobre orientação a objetos em Java:

I. Na classe interface, os métodos são apenas definidos, enquanto na classe abstrata podemos ter métodos apenas definidos e métodos também implementados

II. Na associação de agregação, quando a classe toda é eliminada, elimina-se a classe parte

III. Na associação de composição, a cardinalidade 1 está presente no lado do diamante

Estão corretas

Escolha uma:

a. Apenas III

b. Apenas I e III

c. Apenas I e II

d. Apenas II e III