4.75 + 3.0

7.75

Universidade do Sul de Santa Catarina – UNISUL
Curso de Ciência da Computação
Disciplina: Programação Orientada a Objetos
Professor: Clávison Martinelli Zapelini **E-mail:** clavison.zapelini@unisul.br
AVALIAÇÃO II

Aluno(a): Bernardo José Cardoso Moy

Observações:
A avaliação se encerrará exatamente às 22:00 (sem intervalo). Deve ser executada de forma individual e sem consulta. As dúvidas em relação às questões serão esclarecidas nos momentos iniciais juntamente com a leitura da prova.

Utilize as Classes: Curso, Graduacao, PosGraduacao, Especializacao, Mestrado, Doutorado e Aluno para responder todas as questões da prova

```
public class Curso {
  private String nome;
  private int vagas = 1;

public Curso(String nome){
      this.nome = nome;
}

public void fazMatricula(Aluno a){
      if(getVagas() > 0){
            vagas --;
            a.setCurso(this);
      }
}

public String toString() {
      return getNome()+"-"+getVagas();
}

      //GETS E SETS IMPLEMENTADOS
}
```

```
public class Graduacao extends Curso {

public Graduacao(String nome) {
      super(nome);
}

public String toString() {
 return super.toString()+"
(Graduação)";
}
}
```

```
public class PosGraduacao extends Curso {

public PosGraduacao(String nome) {
      super(nome);
}

public String toString() {
      return super.toString()+" (PG)";
}

}
```

```
public class Mestrado extends PosGraduacao
{
public Mestrado(String nome) {
      super(nome);
}

public void fazMatricula(Aluno a) {
      if(a.getNivel() >= 1)
            super.fazMatricula(a);
      }

public String toString() {
      return super.toString()+":
Mestrado";
}

}
```

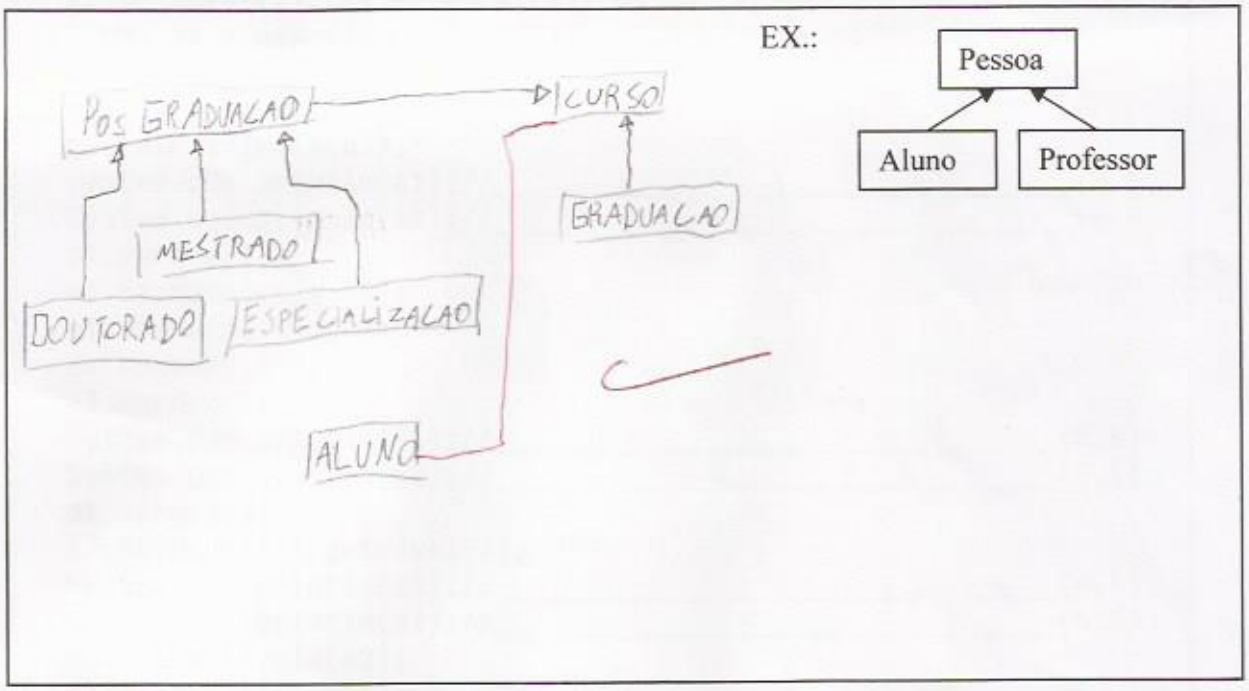
```
public class Especializacao extends
PosGraduacao {

public Especializacao(String nome) {
      super(nome);
}

public void fazMatricula(Aluno a) {
      if(a.getNivel() > 0)
            super.fazMatricula(a);
}

public String toString() {
      return super.toString()+": Espec";
}

}
```

Entregue - 2 laços

```java
public class Doutorado extends
PosGraduacao {

public Doutorado(String nome) {
	super(nome);
}

public void fazMatricula(Aluno a) {
	if(a.getNivel() >= 2)
		super.fazMatricula(a);
}

public String toString() {
	return super.toString()+":
Doutorado";
}
```

```java
public class Aluno {

private String nome;
private Curso curso;
private int nivel = 0

public Aluno(String nome){
	this.nome = nome;
}

public void aprova(){
	nivel ++;
}

public String toString() {
	return getNome()+" - "+curso+" -
"+getNivel();
}
	//GETS E SETS IMPLEMENTADOS

}
```

0.5 Questão 1: Desenhe o diagrama de classes com os devidos relacionamentos que representem a hierarquia de herança utilizada nas classes especificadas para a avaliação. Utilize apenas o nome da classe e o relacionamento conforme o exemplo (0.5 pontos):



ANA - G1 - 0 (GRADUAÇÃO)

Identifique o que será escrito na tela em cada linha destacada nas questões 2, 3 e 4.

0.75

Questão 2:

```
public static void main(String[] args) {
  Aluno a = new Aluno("Ana");
  Graduacao g = new Graduacao("G1");   g.setVagas(10);
  Doutorado d = new Doutorado("D1");   d.setVagas(3);
  g.fazMatricula(a);
  System.out.println(a);//_____4_____(0.5)
  a.aprova();
  g.fazMatricula(a);
  a.aprova();
  System.out.println(a);//_____4_____(0.5)
  d.fazMatricula(a);
  System.out.println(a);//_____4_____(0.5)
}
```

1.0

Questão 3:

```
public static void main(String[] args) {
  Curso g1 = new Graduacao("G1");
  Curso m1 = new Mestrado("M1");
  Curso d1 = new Doutorado("D1");
  g1.setVagas(2); m1.setVagas(2); d1.setVagas(1);
  Aluno a1 = new Aluno("Pedro");
  Aluno a2 = new Aluno("Maria");
  g1.fazMatricula(a1);
  g1.fazMatricula(a2);
  System.out.println(a1);//_____4_____(0.5)
  System.out.println(a2);//_____4_____(0.5)
  a1.aprova();
  m1.fazMatricula(a1);
  m1.fazMatricula(a2);
  a1.aprova();
  a2.aprova();
  System.out.println(a1);//_____4_____(0.5)
  System.out.println(a2);//_____X_____(0.5)
  a1.aprova();
  a2.setNivel(a1.getNivel());
  System.out.println(a1);//_____4_____(0.5)
  System.out.println(a2);//_____X_____(0.5)
  d1.fazMatricula(a2);
  d1.fazMatricula(a1);
  System.out.println(a1);//_____X_____(0.5)
  System.out.println(a2);//_____X_____(0.5)
}
```

1.9

Questão 4:

```
public static void main(String[] args) {

ArrayList<Curso> lc = new ArrayList<Curso>();
lc.add(new Graduacao("G1"));
lc.add(new Mestrado("M1"));
lc.add(new Doutorado("d1"));
Aluno a1 = new Aluno("João");
lc.get(0).fazMatricula(a1);
a1.aprova();
System.out.println(a1);//  JOÃO - G1 - 1 ✓          (0.25)

System.out.println(lc.get(0));//  G1 - 0 (GRADUAÇÃO) ✓  (0.25)

System.out.println(lc.get(1));//  M1 - 1: MESTRADO ✓   (0.25)

System.out.println(lc.get(2));//  d1 - 1: DOUTORADO ✓  (0.25)
lc.get(1).fazMatricula(a1);
a1.aprova();
System.out.println(a1);//  JOÃO - M1 - 2 ✓          (0.25)

System.out.println(lc.get(0));//  G1 - 0 (GRADUAÇÃO) ✓  (0.25)

System.out.println(lc.get(1));//  M1 - 0: MESTRADO ✓   (0.25)

System.out.println(lc.get(2));//  d1 - 1: DOUTORADO ✓  (0.25)
lc.get(2).fazMatricula(a1);
a1.aprova();
System.out.println(a1);//  JOÃO - d1 - 3 ✓          (0.25)

System.out.println(lc.get(0));//  G1 - 0 (GRADUAÇÃO) ✓  (0.25)

System.out.println(lc.get(1));//  M1 - 0: MESTRADO ✓   (0.25)

System.out.println(lc.get(2));//  d1 - 0: DOUTORADO ✓  (0.25)
}
```

0.5

Questão 5: De acordo com os conceitos de relacionamentos entre classes responda qual o tipo de relacionamento que existe entre as classes e justifique sua resposta (1 ponto):

a) Curso e Aluno: Associação, pois é necessário se ter um curso para se ter um aluno. ✓

b) PosGraduacao e Mestrado: Composição, pois o mestrado é como se fosse um nível para chegar ao doutorado, assim o mestrado está compondo a pós graduação.

agregação
associação TEM-UM ✗
composição