8.75

7.25 + 1.5

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): Tiago Bolina

Observações:
A avaliação se encerrará exatamente ás 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
}
```

OK

1 vaga

1 - 1 = 0

G1

10 - 1 = 9

```
public class Graduacao extends Curso {

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

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

OK

G1 (Graduação)

```
public class PosGraduacao extends Curso {

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

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

}
```

OK

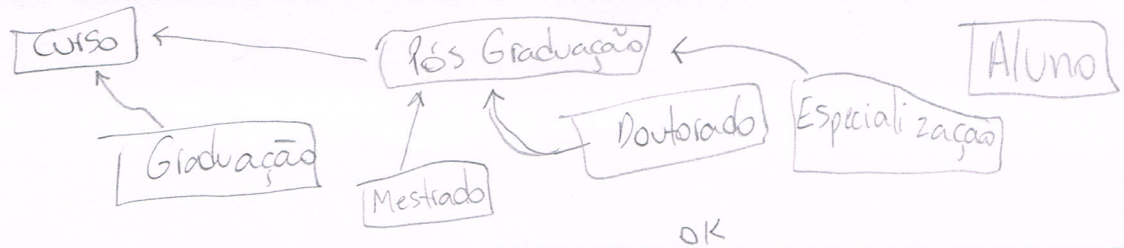
```
public class Mestrado extends PosGraduacao
{
public Mestrado(String nome) {
      super(nome);
}

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

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

}
```

OK

Mestrado (PG)

```
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";
}

}
```

OK



ok

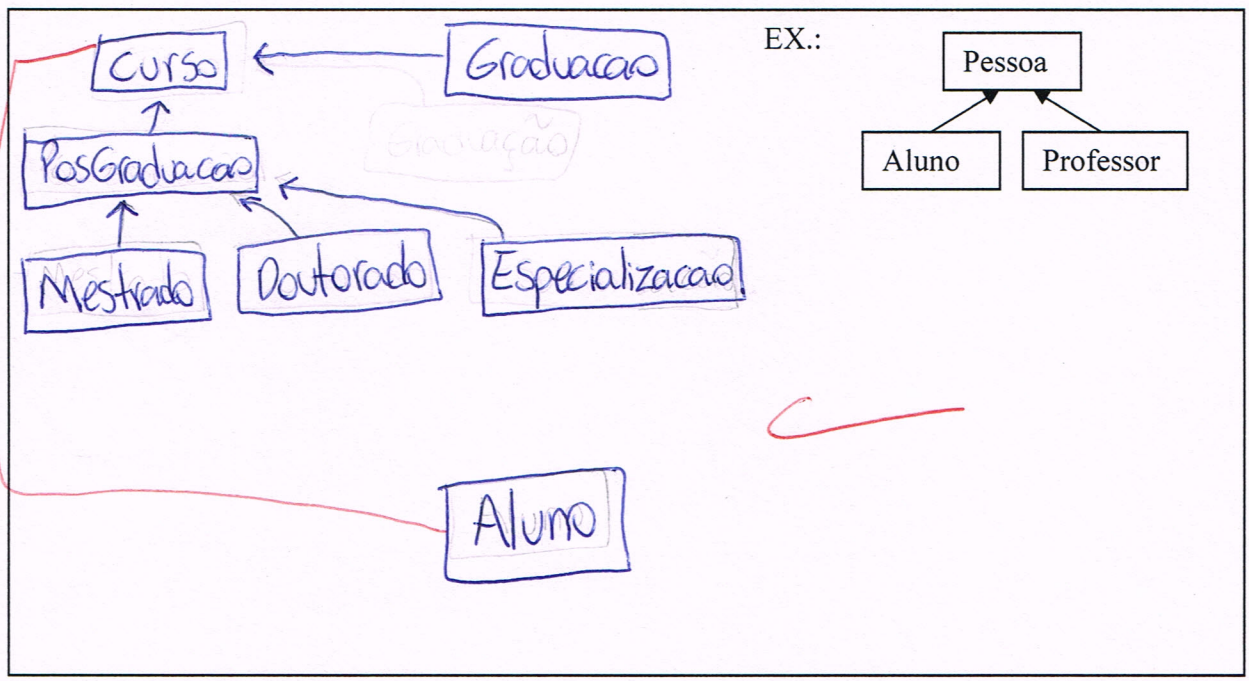
```
public class Doutorado extends
PosGraduacao {

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";
}
```

OK

```
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):



Doutorado → 3 vagas

Ana → nível 2

Aluno = Nome Al - Curso - nível

1.25

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

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);//
  a.aprova();
  g.fazMatricula(a);
  a.aprova();
  System.out.println(a);//
  d.fazMatricula(a);
  System.out.println(a);//
}
```

Ana - G1 (Graduação) - 9 - 0 (0.5)

Ana - G1 - 1 (nível)

Ana - (Graduação) - 8 - 1

nível ++ = 2

Ana - G1 (Graduação) - 8 - 2 (0.5)

Ana - D1: Doutorado (PG) - 2 - 2 (0.5)

Questão 3:

1.5

```
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);//
  System.out.println(a2);//
  a1.aprova();
  m1.fazMatricula(a1);
  m1.fazMatricula(a2);
  a1.aprova();
  a2.aprova();
  System.out.println(a1);//
  System.out.println(a2);//
  a1.aprova();
  a2.setNivel(a1.getNivel());
  System.out.println(a1);//
  System.out.println(a2);//
  d1.fazMatricula(a2);
  d1.fazMatricula(a1);
  System.out.println(a1);//
  System.out.println(a2);//
}
```

vagas

G1 - 1 (Graduação)

G1 - 0 (Graduação)

Pedro - G1 (Graduação) - 0 - 0 (0.5)

Maria - G1 (Graduação) - 0 - 0 (0.5)

nível 1

Pedro - G1 (Graduação) - 0 - 1 (0.5) ✗

Maria - G1 (Graduação) - 0 - 1 (0.5)

Pedro - nível 2

Maria - nível 2

Pedro - G1 (Graduação) - 0 - 2 (0.5) ✗

Maria - G1 (Graduação) - 0 - 2 (0.5) ✗

Maria - Doutorado

Pedro - G1 (Graduação) - 0 - 2 (0.5) ✗

Maria - D1: Doutorado (PG) - 0 - 2 (0.5) ✗

→ Pedro não tem vaga p/ Doutorado

3.0

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(Graduação) - 0 - 1          (0.25)

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

System.out.println(lc.get(1));// M1 : Mestrado (PG) - 1        (0.25)

System.out.println(lc.get(2));// d1 : Doutorado (PG) - 1       (0.25)
lc.get(1).fazMatricula(a1);
a1.aprova();
System.out.println(a1);//João - M1 : Mestrado (PG) - 0 - 2     (0.25)

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

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

System.out.println(lc.get(2));// d1 : Doutorado (PG) - 1       (0.25)
lc.get(2).fazMatricula(a1);
a1.aprova();
System.out.println(a1);//João - d1 : Doutorado (PG) - 0 - 3    (0.25)

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

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

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

1.0

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: Um aluno TEM-UM curso, fica claro em private Curso curso;

b) PosGraduacao e Mestrado: Mestrado É-UMA pós graduação, como se pode ver em extends PosGraduacao

1.25 5 – De acordo com os conceitos de Programação Orientada a Objetos, o programa abaixo possui 4 erros que impedem a sua execução. Identifique os erros NA ORDEM EM QUE APARECE NO PROGRAMA.
Explique o motivo do erro:

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

        Banda b1 = new Banda(" "," ",0);
        b1.getInstrumentos().add(new Instrumento());
        b1.getInstrumentos().add(new Instrumento("A","B"));
        Banda b2 = new Banda();
        System.out.println(b1.toString());
        Instrumento i1 = new Instrumento(" ");
        i1.nome("Flauta");
        i1.tipo("Sopro");
        b2.add(new Instrumento(0));
        System.out.println(i1);

    }
```

Erro 1: Banda b2 = new Banda(); não existe construtor implementado para Banda();

Erro 2: i1.nome("flauta"); erro de sintaxe o correto seria i1.setNome("flauta");

Erro 3: i1.Tipo("Sopro"); erro de sintaxe o correto seria: i1.setTipo("Sopro");

Erro 4: b2.add(new Instrumento(0)); erro de sintaxe o correto seria b2.getInstrumentos().add(i1);