

## PRÁTICA 5

**Aluno (a):**

Questão 1) Leia as sentenças abaixo e assinale todas as VERDADEIRAS:

- (F) Métodos estáticos podem ser abstratos.
- (F) Construtores podem ser abstratos.
- (V) Classes abstratas podem ter construtores.
- (F) Métodos abstratos podem ser privativos.
- (V) Uma classe abstrata podem estender uma normal.
- (V) Posso ter uma classe abstrata sem nenhum método abstrato.

Questão 2) Sobre as classes abstratas puras, assinale a alternativa correta:

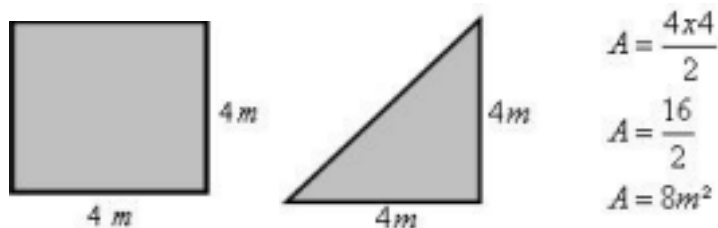
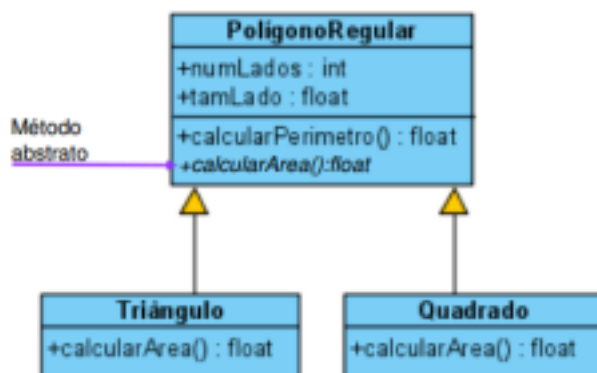
- X) Uma classe abstrata é pura quando possui métodos abstratos; Não possui métodos concretos; e Não possui atributos (não-static).
- b) Uma classe abstrata é pura quando possui métodos abstratos e públicos; Pode possuir métodos concretos e abstratos; e Não possui atributos (não-static).
- c) Uma interface é uma classe abstrata pura; Pode possuir métodos concretos e abstratos.
- d) Uma interface é uma classe abstrata pura; Pode possuir métodos e atributos concretos e abstratos.
- e) Uma classe abstrata é pura quando possui métodos abstratos e públicos; É denominada Interface; Pode possuir métodos concretos e abstratos.

Questão 3) Suponha que você queira estabelecer uma relação de herança entre as classes A e B, onde A é superclasse de B. Deste modo, qual das alternativas abaixo É CORRETA?

- X) `class B extends A {}`
- b) `class B inherits class A {}`
- c) `class B extends class A {}`
- d) `class B + class A {}`
- e) `class A extends class B {}`

Questão 4) Analise o Diagrama de Classe abaixo e escreva o código do programa em Java.

OBS: É necessário fazer o Main, instanciar os objetos Triângulo e Quadrado e chamar o método calcularArea() de cada um.



Área do quadrado: 4 m x 4 m = 16 m<sup>2</sup>

Questão 5) O que será exibido pelo programa abaixo?

```

public class A {
    public int i;
    void display() {
        System.out.println(i);
    }
}

public class B extends A {
    public int j;
    void display() {
        System.out.println(j);
    }
}

public class questaoP001 {

    public static void main(String[] args) {
        B obj = new B();
        obj.i=1;
        obj.j=2;
    }
}
  
```

```
obj.display();  
}  
}
```

2



- a) 1.
- X) 2.
- c) 0.
- d) Nada, pois há um erro de execução.
- e) Nada, pois há um erro de compilação.

Questão 6) Analise o código fonte abaixo, comente as classes e mostre o que irá aparecer na saída do main(). (3,5 pontos)

```
1 package br.com.prova00;  
2 import java.text.SimpleDateFormat;  
3 import java.util.Calendar;  
4 import java.util.Date;  
5  
6 public abstract class Convite {  
7     String timeStamp;  
8     String usuario;  
9     String mensagem;  
10  
11     public Convite(String usuario, String mensagem) {  
12         timeStamp = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss").format(Calendar.getInstance().getTime());  
13         this.usuario = usuario;  
14         this.mensagem = mensagem;  
15     }  
16     public String getUsuario() {  
17         return usuario;  
18     }  
19  
20     public void setUsuario(String usuario) {  
21         this.usuario = usuario;  
22     }  
23  
24     public String getMensagem() {  
25         return mensagem;  
26     }  
27  
28     public void setMensagem(String mensagem) {  
29         this.mensagem = mensagem;  
30     }  
31  
32     public String getdataEnvio() {  
33         return timeStamp;  
34     }  
35     public abstract String mostrar();  
36  
37 }  
38
```

3



```

1 package br.com.prova00;
2
3 public class convidarEmail extends Convite{
4     String email;
5
6     public convidarEmail(String usuario, String mensagem, String email) {
7         super(usuario, mensagem);
8         this.email = email;
9     }
10
11     public String getEmail() {
12         return email;
13     }
14
15     public void setEmail(String email) {
16         this.email = email;
17     }
18
19     public String mostrar() {
20         return "Usuário: " + getUsuario() + "\nData Envio: " + getdataEnvio() + "\nMensagem: " + getMensagem() + "\nE-mail: " + getEmail() ;
21     }
22 }

```

---

```

1 package br.com.prova00;
2
3 public final class convidarWhatsApp extends Convite{
4     String telefone;
5
6
7     public convidarWhatsApp(String usuario, String mensagem, String telefone) {
8         super(usuario, mensagem);
9         this.telefone = telefone;
10    }
11
12    public String getTelefone() {
13        return telefone;
14    }
15
16    public void setTelefone(String telefone) {
17        this.telefone = telefone;
18    }
19
20    public String mostrar() {
21        return "Usuário: " + getUsuario() + "\nData Envio: " + getdataEnvio() + "\nMensagem: " + getMensagem() + "\nTelefone: " + getTelefone() ;
22    }
23
24 }

```

```
package br.com.prova00;
```

```
public class EnvioConvite {
```

```

    public static void main(String[] args) {
        convidarWhatsApp conviteWhats = new
        convidarWhatsApp("Michelle", "Olá, hoje tem prova", "(31)
        98543-0198"); System.out.println(conviteWhats.mostrar());
        System.out.println("*****");
        convidarEmail conviteEmail = new convidarEmail("Michelle", "Olá,
        hoje tem prova", "michellehanne.andrade@gmail.com");
        System.out.println(conviteEmail.mostrar());
        System.out.println("*****");
        Convite convite = new Convite("Michelle", "Olá, hoje tem prova");
        System.out.println(Convite.mostrar());
    }
}

```

O código não irá compilar porque a linha `Convite convite` está tentando instanciar uma classe abstrata.