**PRÁTICA 5**

**Aluno (a): Roger William Roque da Silva**

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:

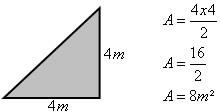
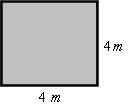
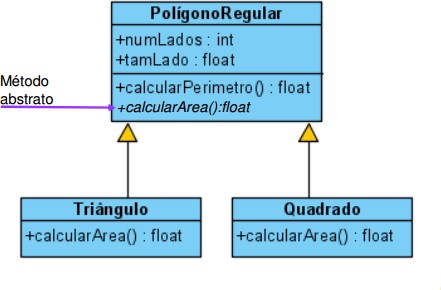
1. **Uma classe abstrata é pura quando possui métodos abstratos; Não possui métodos concretos; e Não possui atributos (não-static).**
2. Uma classe abstrata é pura quando possui métodos abstratos e públicos; Pode possuir métodos concretos e abstratos; e Não possui atributos (não-static).
3. Uma interface é uma classe abstrata pura; Pode possuir métodos concretos e abstratos.
4. Uma interface é uma classe abstrata pura; Pode possuir métodos e atributos concretos e abstratos.
5. 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?

1. **class B extends A {}**
2. class B inherits class A {}
3. class B extends class A {}
4. class B + class A {}
5. 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 m²

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** questaoPOO1 {

**public static void** main(String[] args) { B obj = **new** B();

obj.i=1; obj.j=2; obj.display();

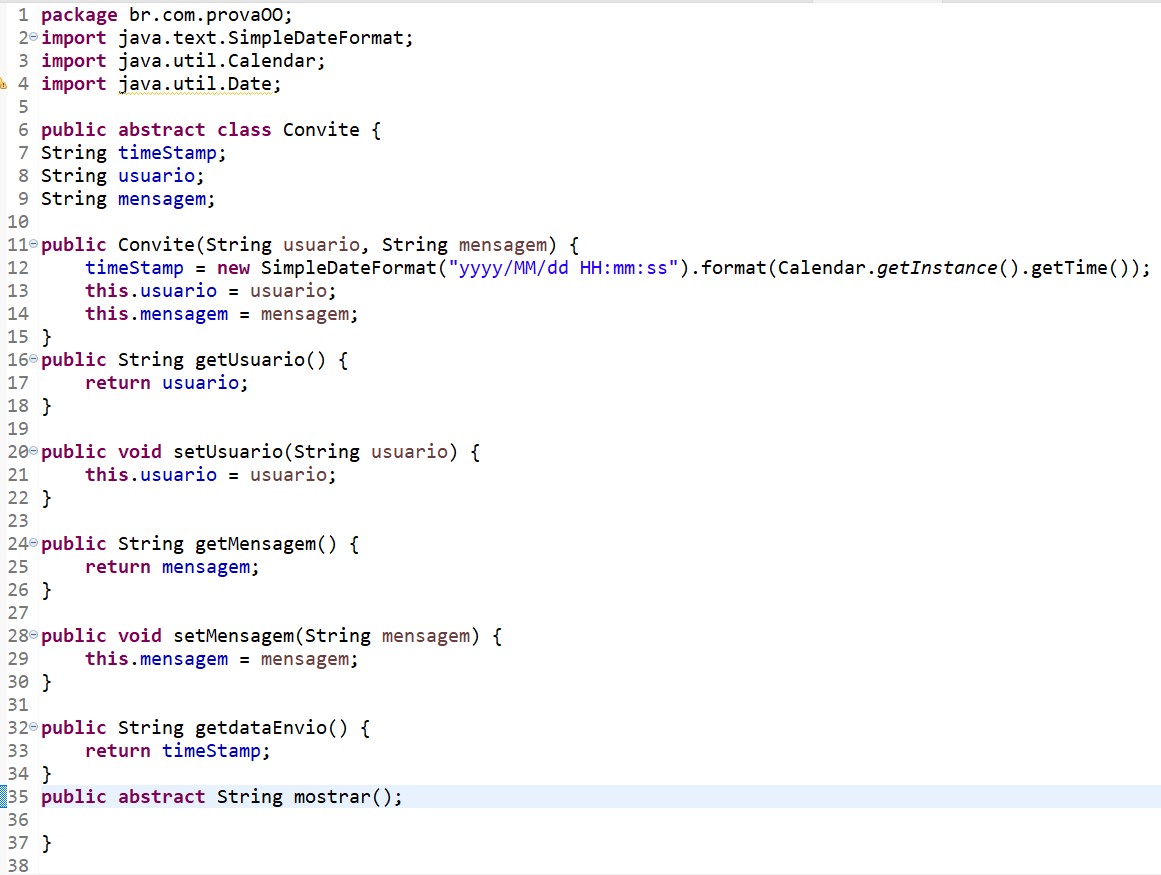
}

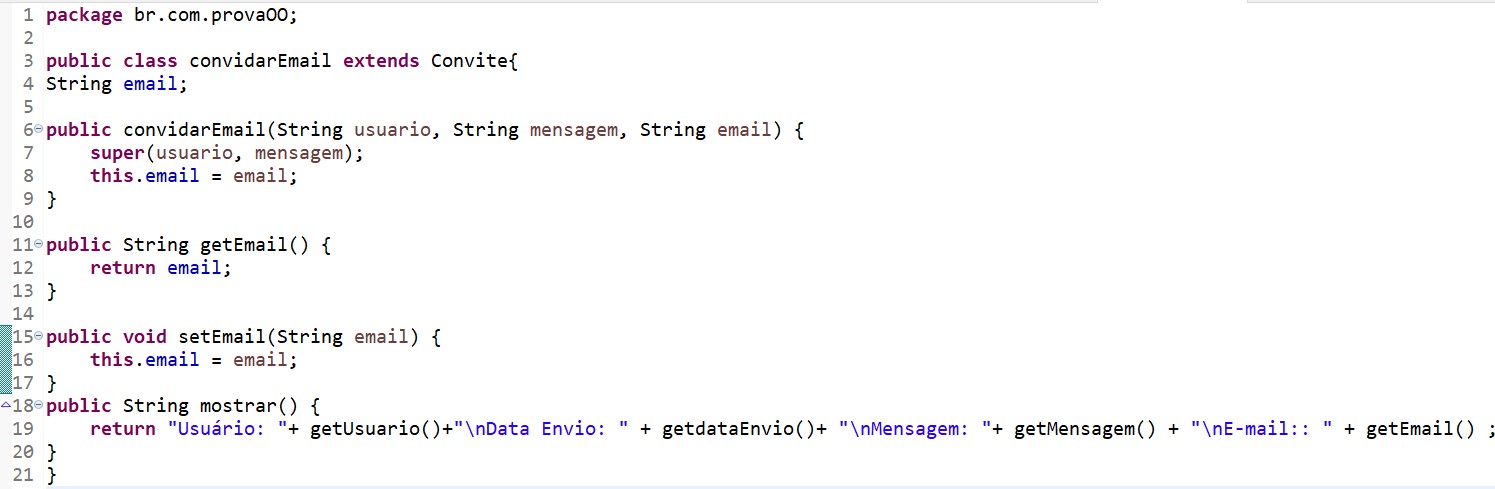
}

* 1. 1.
  2. **2.**
  3. 0.
  4. Nada, pois há um erro de execução.
  5. 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)

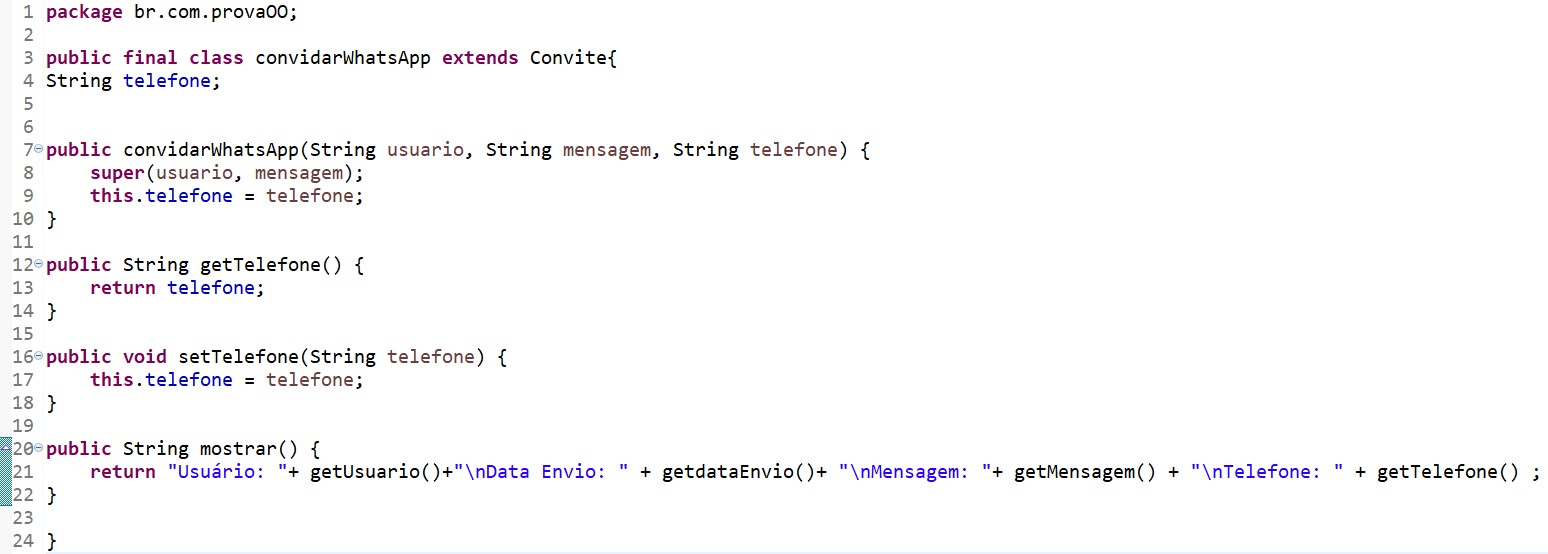
**Class abstrata Convite que possui 3 Strings: timeStamp, usuario e mensagem.**





**Class convidarEmail filha de convite que possui String email; retornando a string mostrar**

**Class convidarWhatsapp filha de convite que possui a string telefone; retornando a string mostrar**



**Class EnvioConvite e o main mostra os retornos dos atributos concatenados através dos Gets.**

**package** br.com.provaOO;

**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](mailto: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());

}

}