PRÁTICA 5

**Aluno: Wadson Augusto Dos Santos Rocha  
 RA: 12115920**

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).



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

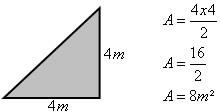
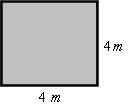
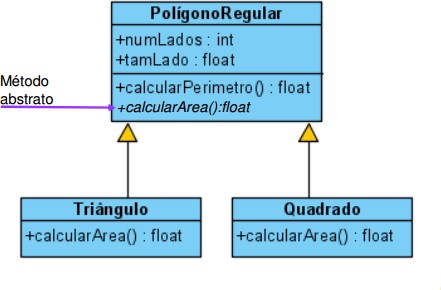
1. class B extends A {}



1. class B inherits class A {}
2. class B extends class A {}
3. class B + class A {}
4. 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:

**Código:**

public class Main {

public static void main(String[] args) {

Triangulo triangulo = new Triangulo(3, 4);

Quadrado quadrado = new Quadrado(4);

System.out.println("Área do triângulo: " + triangulo.calcularArea() + " m²");

System.out.println("Área do quadrado: " + quadrado.calcularArea() + " m²");

}

}

class Triangulo {

private double base;

private double altura;

public Triangulo(double base, double altura) {

this.base = base;

this.altura = altura;

}

public double calcularArea() {

return (base \* altura) / 2;

}

}

class Quadrado {

private double lado;

public Quadrado(double lado) {

this.lado = lado;

}

public double calcularArea() {

return lado \* lado;

}

}

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();

}

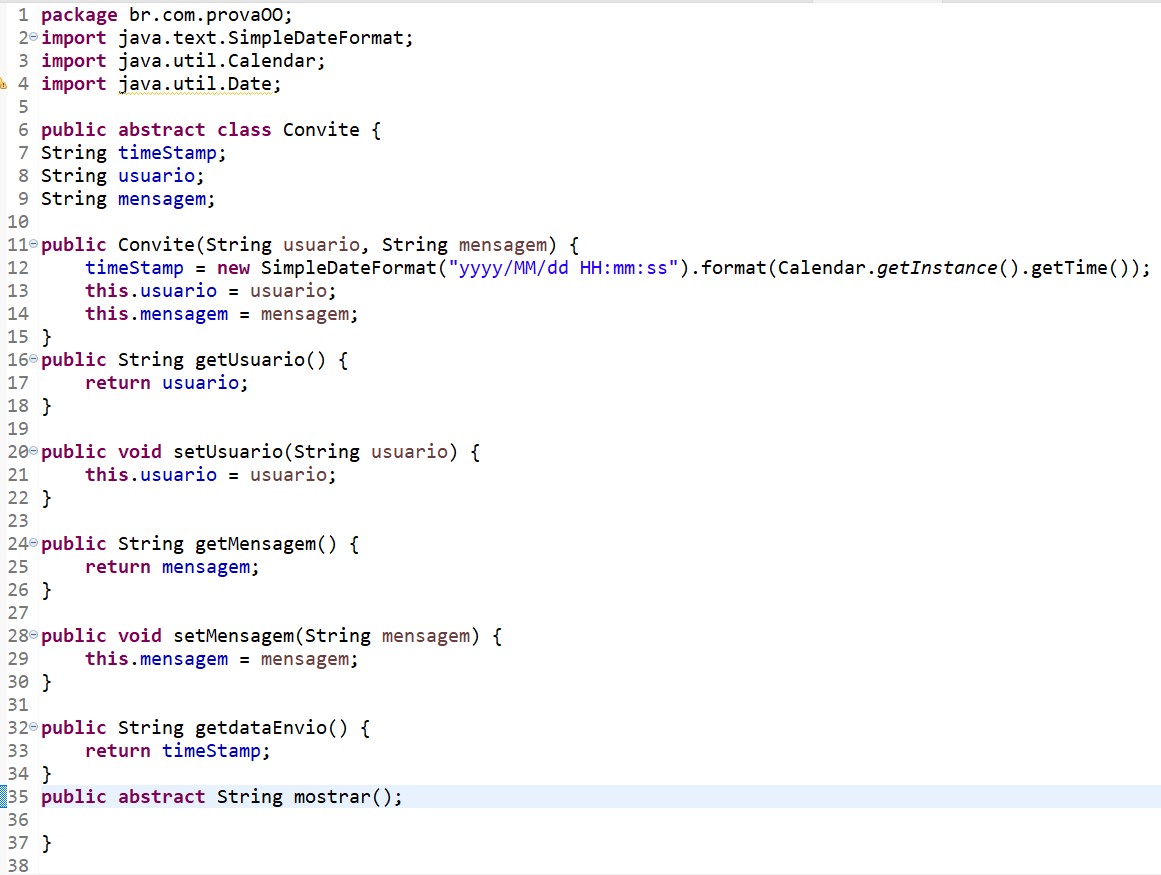
}

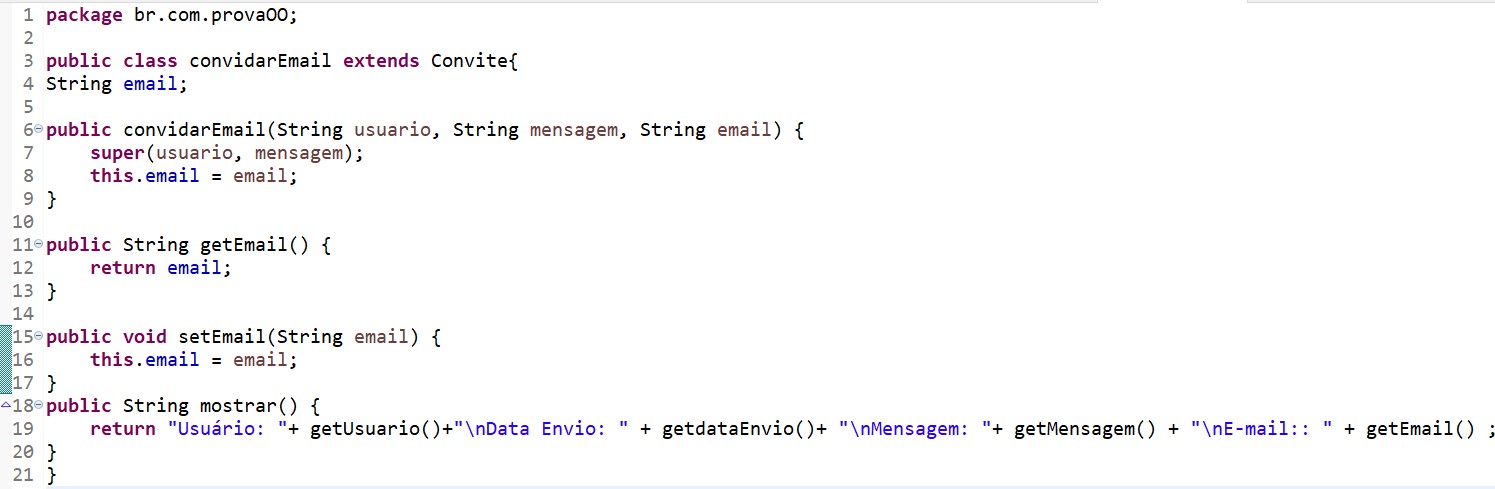
* 1. 1.
  2. 2.

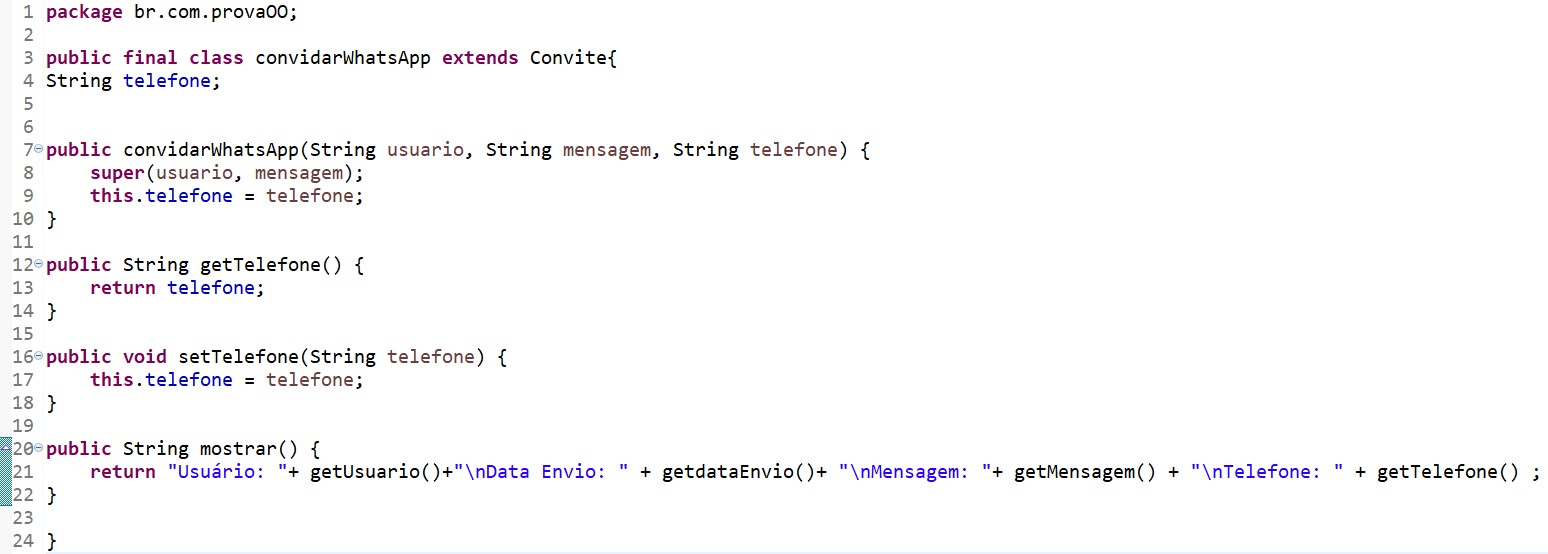


* 1. 0.
  2. Nada, pois há um erro de execução.
  3. Nada, pois há um erro de compilação.

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







**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());

}

}

**Resposta:**

A classe Convite é a classe base para os convites e define um convite genérico. Ela possui um construtor que recebe dois parâmetros: o nome do destinatário e a mensagem do convite. Além disso, ela possui um método mostrar que retorna uma string com a mensagem formatada do convite.

A classe convidarEmail é uma subclasse de Convite e define um convite que pode ser enviado por email. Ela possui um construtor que recebe três parâmetros: o nome do destinatário, a mensagem e o endereço de email para enviar o convite. Além disso, ela possui um método mostrar que retorna uma string com a mensagem formatada do convite.

A classe convidarWhatsApp é uma subclasse de Convite e define um convite que pode ser enviado pelo WhatsApp. Ela possui um construtor que recebe três parâmetros: o nome do destinatário, a mensagem e o número de telefone para enviar o convite. Além disso, ela possui um método mostrar que retorna uma string com a mensagem formatada do convite.

No método main, o programa cria três objetos: um convidarWhatsApp, um convidarEmail e um Convite. Em seguida, ele chama o método mostrar para cada um deles e imprime a saída no console:

Convite via WhatsApp:

Olá, Michelle! Você está convidado(a) para a prova hoje.

Por favor, confirme sua presença.

Telefone para contato: (31) 98543-0198

Convite por email:

Olá, Michelle! Você está convidado(a) para a prova hoje.

Por favor, confirme sua presença.

Endereço de email para contato: michellehanne.andrade@gmail.com