Teste AC 1 – Polimorfismo

Usa-se o Resposta [o polimorfismo] ao substituir um método de Resposta [superclasse] numa Resposta [subclasse] criando um Resposta [método] com o mesmo nome e lista de parâmetros.

Usar o mesmo nome de método para indicar diferentes implementações é chamado

Resposta correta: polimorfismo

Na procura do método a executar (*Method lookup*), para dar prioridade aos métodos redefinidos, usamos a anotação Resposta [Override].

Considere as seguintes classes:

```
class Animal {
  public String makeSound() {
    return "não produz som"; }
}

class Cat extends Animal {
  public String makeSound() {
    return "Miau"; }
}

class StreetCat extends Cat {
}

class AmericanBobtail extends Cat {
 public String makeSound() {
    return "Miauau"; }
}
```

Considere agora o seguinte código no programa principal:

```
ArrayList<Animal> animals = new ArrayList<>();

animals.add(new Cat());
animals.add(new Animal());
animals.add(new AmericanBobtail());

for( Animal a : animals )
  System.out.println( a.makeSound());
```

O que é mostrado no ecrã quando o programa principal é corrido?

Resposta correta: `Miau não produz som Miauau`

Considerando que a classe `Document` é uma generalização da classe `Magazine`.

A seguinte instrução

```
Document d = new Magazine("Visão",34);
```

está:

Resposta correta: Correta, pois em Java existe o princípio da substituição.

Teste AC 2 – Classes Abstratas e Interfases

A diferença entre uma interface e uma classe abstracta é que...

Resposta correta: a interface não permite a definição de métodos e a classe abstracta permite

Considere o seguinte código em JAVA

```
public interface Cleaner{public void clean();}
```

E assuma que a classe `Vacuumcleaner` implementa a interface `Cleaner`.

Qual dos seguintes conjuntos de instruções é **correcto**:

Resposta correta: `Cleaner[] list= new Cleaner[100];`

Considerando que a classe `Car` é abstracta, qual dos seguintes conjuntos de instruções está **correcto:**

Resposta correta: `Car[] listC = new Car[2];`

Considere que existem as interfaces `Movable` e `Drawable`.

Qual das seguintes instruções em JAVA se pode considerar **errada**:

Resposta correta: `public class Drawing extends Drawable {...`

A palavra reservada `abstract` é usada para definir:

Resposta correta: classe abstratas e métodos abstratos

Qual a afirmação **correta**?

Resposta correta: Um método abstracto não possui código.

Considere o seguinte código em JAVA

```
public interface A {
   public int metodoA (int valor);
}

public interface B {
   public String metodoB (String valor);
}

public interface C extends A, B
   { //código }
```

Qual das seguintes afirmações está **errada:**

Resposta correta: O código está errado, deveria ser:
`public interface C implements A, B { //código }`

Considerando o seguinte código:

```
interface Censurable { String getCritic(); }

class Referee implements Censurable {

public String getCritic() { ... }

public void whisle() { ... }

... }

Censurable person = new Referee();
```

Qual das seguintes linhas está **correcta:**

Resposta correta: `System.out.println( person.getCritic());`

Considere o seguinte codigo em JAVA

```
public interface Washable
  { //código }

public abstract class Vehicle implements Washable
  { //código }

public class Bike extends Vehicle
  { //código }

public class Car extends Vehicle
  { //código }
```

Qual das seguintes afirmações está **correcta:**

Resposta correta: `Car` é uma instância de `Washable`

Considere o seguinte código em JAVA

```
public interface Scalable {
  public void scale(float factor);
}
```

E assuma que a classe `Shape` implementa a interface `Scalable`.

Identifique a afirmação **correcta**

Resposta correta: A classe `Shape` se não for abstrata e não herdar de nenhuma outra classe tem que obrigatoriamente implementar o método `scale`

Numa hierarquia de classes uma classe base abstracta é normalmente usada para:

Resposta correta: declarar métodos que devem ser implementados nas classes derivadas

Teste AC 3 – Desenvolvimento de aplicações

Considere as seguintes classes:

```
1  public class Person {
2  private String name,id;
3
4  public Person(String name, String id) {
5  this.name = name;
6  this.id = id;
7  }  }
8
9  public class Pet {
10   private String name;
11   private Person owner;
12   private Adress ownerAdress;
13   private int age;
14
15   public Pet(String name, int age) {
16     this.name = name;
17     this.age = age;
18     owner=null;}
19
20   public void setOwnerName(String name) {
21     owner.setName(name);}
22
23   public int getAge() {
24     return age;}
25
26   public void incrementAge() {
27     age++;}
28
29   public String getName() {
30     return nome;}
31
32   public void setName(String name) {
33     this.name = name;}
34 }
```

Qual dos seguintes problemas acha que deverá existir?

Resposta correta: alto acoplamento de identidades

Considere a seguinte classe:

```
public class Calculator
   {
      private int number1;
      private int number2;
      // ...
      public int add () { ... }
      public int substract () { ... }
      public int divide() { ... }
      public int multiply() { ... }

   }
public class Window {
      public int input() { ... }
      public void displayResult(int result) { ... }
      public void displayError(int result) { ... }
}
```

Qual das seguintes situações acha que deverá ocorrer?

Resposta correta: forte coesão de classes

A Redundância dificulta a actualização dos dados, assim, normalmente, diminuíndo a coesão das entidades envolvidas diminuímos a redundância.

Resposta correta: Falso

Considere a existencia da classe Documento e Processo

Se considerarmos que Processo herda de Documento...

Selecione uma opção:

a. Todas as outras respostas estão incorretas
b. é um erro pois estamos a criar acoplamento de identidades entre as duas classes
c. está correcto, pois estamos a permitir a reutilização de codigo atraves do uso da herança.
d. é um erro pois estamos a criar acoplamento de representação entre classes

Resposta correta: Todas as outras respostas estão incorretas

Considere o esboço de cartas CRC apresentado a seguir para uma loja online.

1	
Faz encomendas Conhece o nome Conhece o enderaço Conhece o número de cliente Conhece o histórico de encomendas	2

3	
Conhece posicionamento da data Conhece a data de entrega Conhece o total Conhece as taxas a aplicar Conhece o número de encomenda Conhece a ordem dos ítens	4

Identifique as classes e responsabilidades 1, 2 , 3 e 4

A sua resposta está correta.

A resposta correcta é: 3 → Encomenda, 1 → Cliente, 2 → Encomenda, 4 → Item

Pediu-se a um aluno para construir um programa para uma agência de viagens. O aluno identificou uma classe `TravelKit` onde incluia um objecto da classe `Client` com os dados do cliente. No mesmo programa, dentro da classe `Client` colocou um array de objectos da classe `TravelKit` que continha todas as viagens que o cliente tinha adquirido. Se o modelo proposto pelo aluno fosse implementado que problemas encontrava?

Resposta correta: Um alto acoplamento.

Classes e Métodos pouco coesos...

Respostas corretas: Dificultam a reutilização do código., dificultam a adaptação a novos requisitos e a expansão de funcionalidades.

Considere a seguinte classe:

```
1  public class BoardGame {
2  private Piece[] pieces;
3  private Board board;
4  private Room gameRoom;
5  private Date gameDate;
6
7  // métodos...
8
9  }
```

Qual dos seguintes problemas de coesão de classes acha que deverá existir?

Resposta correta: mistura de domínios

Teste AC 4 – Genéricos e Coleções

Considere o seguinte código:

```
HashMap<Integer,Invoice> invoices = new HashMap<>();

invoices.put(1, new Invoice(1) );
invoices.put(2, new Invoice(2) );
```

Sabendo que o número inteiro representa o número da fatura, qual dos seguintes ciclos escolheria se quisesse iterar todos os pares de número de fatura/fatura existentes:

Resposta correta: `for( Map.Entry em : invoices.entrySet() )`

Considere uma colecção que contem os elementos `x` e `y` e em que `x.equals(y)` retorna o valor `"true"`.

Qual o tipo de coleção que poderá estar a representar este grupo de objetos:

Resposta correta: `List`

Um iterador (iterator) serve para operar sobre os elementos que estão dentro de uma colecção um a um. Indique o que falta nos espaços assinalados com _____, respectivamente:

```
Map<Integer,String> students = new HashMap<>();

Iterator<String> it = students.values()._________;

while(it.hasNext()) {

                String s = _________;

                System.out.println(s);

}
```

Selecione uma opção de resposta:

Resposta correta: `iterator()` e `it.next();`

É válido o seguinte código?

```
<HashMap<Integer,String> aMap = new Map<>();
```

Resposta correta: Falso

Considere o código seguinte:

```
public static < ___ > void printArray(____[] elements) {
   for ( ___ element : elements){
        System.out.println(element);
      }
   System.out.println();
    }

public static void main( String args[] ) {
   Integer[] intArray = { 10, 20, 30, 40, 50 };
   Character[] charArray = { 'J', 'A', 'V', 'A',
'T','P','O','I','N','T' };

   System.out.println( "Impressão de um array de inteiros" );
   printArray( intArray  );

   System.out.println( "Impressão de um array de caracteres" );
   printArray( charArray );
}
```

De acordo com a genericidade, que caractere deverá constar no código nas zonas por completar, assinaladas com __

Resposta correta: E

Um ciclo Foreach pode ser usado para iterar sobre uma colecção `ArrayList<String> lista` através de:

Resposta correta: `for(String s: lista)`

Considere o código a seguir:

```
class MyGen<T>{
  T obj;
  void add(T obj){this.obj=obj;}
    T get(){
      return obj;}
}
class App{
public static void main(String args[]){
  MyGen<_____> m=new MyGen<_____>();
  m.add(new Student("João");
  System.out.println(m.get());
}}
```

De acordo com a genericidade, que classe deverá constar no código nas zonas por completar, assinaladas com __ :

Resposta correta: Student

Considere o seguinte código em JAVA:

```
HashSet<Animal> set = new HashSet<Animal>();

  set.add(new Animal("Cao"));

  set.add(new Animal ("Porco"));

  set._________;
```

Assinale TODAS as opções possíveis para completar a ultima instrução de forma a este excerto de código poder ser executado sem dar nenhuma exceção, ou erro de compilação.

Respostas corretas: `size()`, `remove("Porco");`

Considere o seguinte código:

```
Set<String> vals = new HashSet<>();
vals.add("um");
vals.add("dois");
vals.add("três");
vals.add("quatro");
vals.add("um");


Iterator it = vals.iterator();
while (it.hasNext()) {
    System.out.print(it.next() + " ");
}
```

Quais dos seguintes resultados podem ser mostrados no ecrã?

Respostas corretas: dois três quatro um, quatro um três dois, um dois três quatro

Considere o seguinte código:

```
List list = new ArrayList();
list.add("Olà");
String s = (String) list.get(0);

List<String> list = new ArrayList<String>();
list.add("Olà");
String s = list.get(0);
```

Da lista de vantagem associadas á utilização de tipos genéricos apresentada abaixo, qual das vantagens é ilustrada pelo código.

Resposta correta: A conversão de tipo não é necessária: não há necessidade de converter o objeto usando o ***cast***.

Qual deve ser o valor retornado pelo seguinte método?

```
System.out.println("zebra".compareTo("zebrA"));
```

Resposta correta: **um valor inteiro positivo**

A Interface `Set<E>` está na hierarquia da interface `Collection<E>`.

Assim todas as coleções que implementam essa interface possuem os métodos:

Resposta correta: `contains(), isEmpty(), add(), clear()`.

Teste AC 5 – Exceções e Input / Output

Que método da classe `File` é utilizado para testar a existência de um ficheiro ou directoria ?

Resposta correta: `exists()`

Indique a ordem correcta de executar as instruções para escrever no Ficheiro ***Bomdia.txt*** a frase: ***Hoje está sol*** no programa abaixo

```
public class Teste{
  public static void code (String arg[]) {
   <instrucao 1>
   <instrucao 2>
   <instrucao 3>
   <instrucao 4>
   <instrucao 5>
   <instrucao 6>
}
```

A resposta correcta é:

<instrucao 1> → File ficheiro = new File ("Bomdia.txt");

<instrucao 2> → FileWriter file_writer = new FileWriter (ficheiro);

<instrucao 3> → Writer buf_writer = new BufferedWriter (file_writer);

<instrucao 4> → PrintWriter print_writer = new PrintWriter (buf_writer);

<instrucao 5> → print_writer.println ("Hoje esta sol");

<instrucao 6> → print_writer.flush();

Qual o método da classe `Scanner` usado para ler um valor real?

Resposta correta: `nextDouble()`

Qual é o nome da classe abstracta base para canais (*streams)* que lidam com saída de bytes?

Resposta correta: `OutputStream`

Desserializar um objeto pode causar uma `ClassNotFoundException`

Resposta correta: Verdadeiro

Relativo ao código seguinte, complete

```
public class Main {

  public static void main(String[] args){
    Data x = new Data (2012, 15, 31);
    int dia = 0;
    <1> {
      x.setDia(32);
      x.troca(-5,dia);
    }
    <2> (ValorInvalidoException e){
      System.out.println(e.getMessage());
    }
    <3>(TrocaInvalidaException e){
      System.out.println(e.getMessage());
    }
  }
}
```

A resposta correcta é: <1> → try, <2> → catch, <3> → catch

Uma excepção é um evento gerado pelo código do programa portanto pode ser capturado pelo compilador.

Resposta correta: Falso

Tendo em conta o seguinte código de um método:

```
void method()
{
  int a=-12, b=3, res;

  try {
   res = sumPositives( a, b );
  }
  catch(ErrorSumException e)
  {
   System.out.println("Erro: " + e.getMessage() + e.getValor();
   // Resultado no ecrã: "Erro: Valor negativo: -12"
  }
} // metodo
```

Se tivesse de implementar a classe ErrorSumException qual considera que seria o construtor mais adequado dos que se mostram a seguir:

Resposta correta:

```
public ErrorSumException( String message, int valor)
{
 super(message);
 this.valor = valor;
}
```

A diferença entre excepções verificadas e não-verificadas é que:

Resposta correta: as excepções verificadas necessitam de ser capturadas no código enquanto as não-verificadas não necessitam.

Qual das seguintes excepções capturaria no tratamento de um erro provocado por uma divisão por zero?

Resposta correta: `ArithmeticException`

Indique a ordem correcta de executar as instruções para ler do Ficheiro ***file.txt*** um inteiro.

```
public class Teste {

 public static void main (String arg[]) {

  <instrucao 1>

  try{
   <instrucao 2>
   <instrucao 3>
   System.out.println ("Inteiro lido: " + primeiraLinha);

  }

  <instrucao 4>{
    System.out.println ("Mismatch exception:" + e );
  }

   <instrucao 5> {
    System.out.println ("Ficheiro não encontrado!");
    System.exit (0);

  }
 }
}
```

A resposta correcta é:

<instrucao 5> → catch (FileNotFoundException e)

<instrucao 2> → Scanner sc = new Scanner(ficheiro);

<instrucao 1> → File ficheiro = new File ("ficheiro.txt");

<instrucao 4> → catch (InputMismatchException e)

<instrucao 3> → String primeiraLinha = scanner.nextInt();

Teste AC 6 – Introdução a JavaFX

Escolha as afirmações **INCORRETAS**.

a. O método `start()` é o primeiro método a ser executado numa aplicação JavaFX.
b. Para qualquer aplicação JavaFX o programador deve substituir o método `start()`.
c. O método `stop()` sempre é executado após o método `start()`.

Resposta correta: O método `start()` é o primeiro método a ser executado numa aplicação JavaFX.

Complete o seguinte código:

```
{

// ...

Group circles = new Group();

for (int i = 0; i < 30; i++) {

  Circle circle = new Circle(150+2*i, Color.web("white", 0.05));
  circle.setStrokeType(StrokeType.OUTSIDE);
  circle.setStroke(Color.web("white", 0.16));
  circle.setStrokeWidth(4);

  //<< escolha a linha de código >>
}
```

Resposta correta: `circles.getChildren().add(circle);`

Quais das seguintes classes são nós (ou seja, derivadas da classe `Node`)?

Respostas corretas: `Rectangle`, `Text`, `Group`, `Shape`

Considere a interface do utilizador da figura 1.Faça a correspondência entre as letras A até F da figura 2 para identificar as classes da estrutura da aplicação:

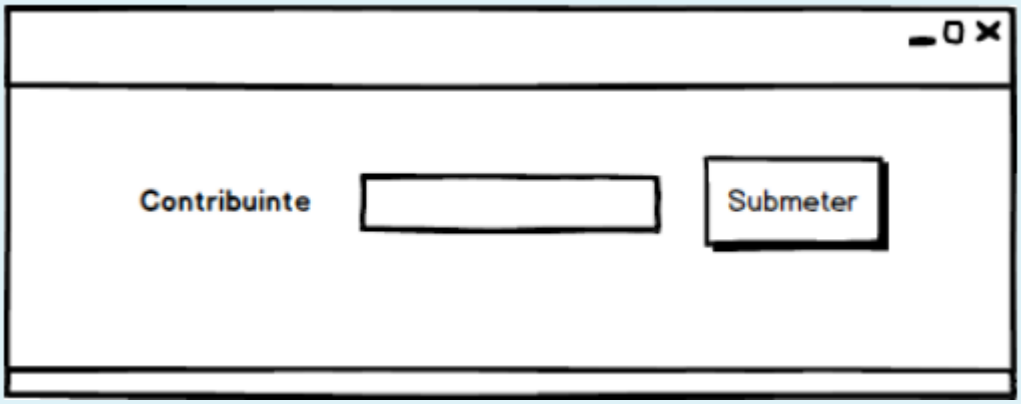


Figura 1

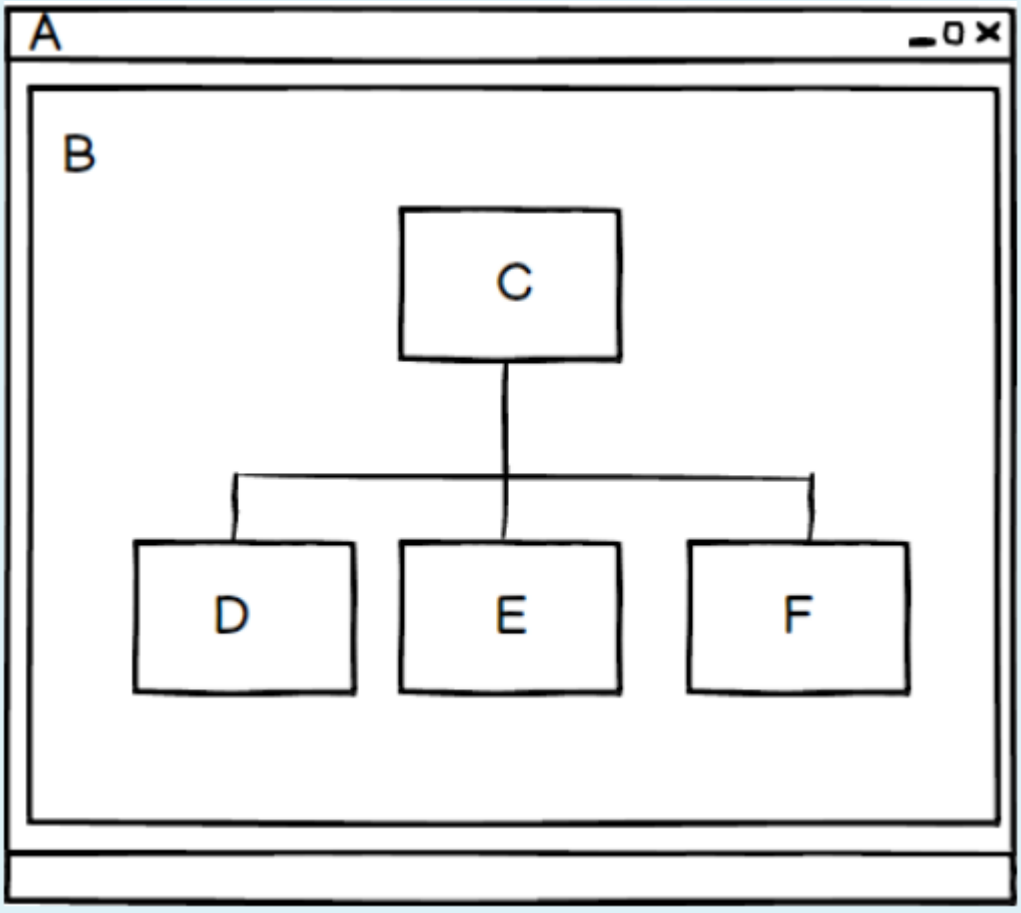


Figura 2

A resposta correcta é: F → Button, B → Scene, A → Stage, C → StackPane, D → Label, E → TextField

Considere o seguinte código de uma aplicação em JavaFX e explique o seu funcionamento:

```
1  @Override
2  public void start(Stage primaryStage) {
3
4    primaryStage.setTitle("Slide 9 Text Fonts");      // <1>
5    Group root = new Group();
6    Scene scene = new Scene(root, 550, 250);
7
8    Text text2 = new Text(50, 50, "Font Serif RED");   // <2>
9    Font serif = Font.font("Serif", 30);              // <3>
10   text2.setFont(serif);
11   text2.setFill(Color.RED);                      // <4>
12   root.getChildren().add(text2);
13
14   primaryStage.setScene(scene);
15   primaryStage.show();
16 }
```

A resposta correcta é:

<1> → Mostra o texto "Slide 9 Text Fonts" na barra de título da janela da aplicação

<2> → Cria um texto com a string "Font Serif RED" que deverá ser mostrada na posição X=50, Y=50

<3> → Cria uma fonte com serifa e com 30 pontos de tamanho

<4> → Altera a cor das letras do texto para vermelho

Considere o seguinte código em JavaFX e preencha corretamente as linhas em falta assinaladas com (1) (2) (3) e (4):

```
(1)

public class HeloWorldMain extends Application {

  public static void main(String[] args) {
    launch(args);
  }

  @Override
  (2)
  {
    primaryStage.setTitle("Hello World!");
    Button btn = new Button();
    btn.setText("Say 'Hello World'");
    btn.setOnAction(new EventHandler<ActionEvent>() {
                  @Override

          (3)

                  {
                 System.out.println("Hello World!");
                 }
             } );

    (4)
    root.getChildren().add(btn);
    primaryStage.setScene(new Scene(root, 300, 250));

  (5)

  }
}
```

A resposta correcta é: (1) → import javafx.stage.Stage;

(2) → public void start(Stage primaryStage) (3) → public void handle(ActionEvent event)

(4) → StackPane root = new StackPane(); (5) → primaryStage.show();

Em JavaFX:

A classe que contém o método `main é subclasse de Application.`

No método `main()` lançamos a aplicação passando os argumentos para o método `Application.launch()`

Quando a aplicação tiver sido inicializada a infraestrutura do javaFX invocará o método << complete aqui >>

Resposta correta: `Application.start()`

Teste AC 7 – JavaFX – Controlos e Eventos

Considere o seguinte código:

```
Button btn = new Button(); btn.setText("Dizer 'Ganda Jogo'"); btn.setOnAction( new EventHandler<ActionEvent>()
    {
    @Override
     public void handle(ActionEvent event) {
            System.out.println("Ganda Jogo");
             }
    });
```

Qual o tipo de acontecimento que deve ocorrer para que o texto "Ganda Jogo" seja mostrado na consola?

Resposta correta: Premida a tecla do rato sobre o botão

Se pretendemos que um conjunto de nós apareçam alinhados horizontalmente devemos usar um contentor do tipo:

Resposta correta: HBox

Ellipse ellipse = EllipseBuilder << 1 >>

.centerX(50)

<< 2 >> //centro na vertical

<< 3 >> //raio na largura

.radiusY(70)

<< 4 >> ;

A resposta correcta é:

<< 1 >> → .create(), << 2 >> → .centerY(100)

<< 3 >> → .radiusX(100), << 4 >> → .build()

Em Java FX qual o método utilizado para alterar o texto que aparece num botão (Button)?

Resposta correta: setText

Pretende-se fazer uma aplicação com o formulário seguinte:

Escolha o painel mais apropriado para conseguir a formatação pretendida.

Resposta correta: GridPane

Pretende-se colocar um botão junto à parte superior da janela e um campo de texto na base da pagina, ambos centrados. Qual o tipo de contentor que deve utilizar.

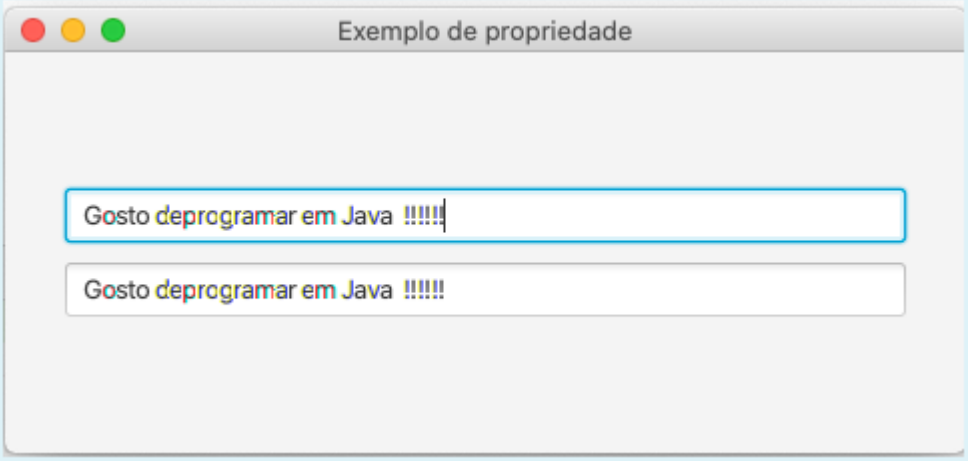
Resposta correta: BorderPane

Teste AC 8 – Propriedades e Controlos Avançados

A classe `ObservableList` é utilizada apenas com controlos de interface do utilizador do tipo `ListView`.

Resposta correta: Falso

Considere o programa composto pela janela abaixo que permite que o texto inserido pelo utilizador no `TextField` de cima aparece em simultâneo no controlo de baixo. Também, se o utilizador insere / remove texto no controlo de baixo, o controlo de cima deve ser atualizado



Considere, também, o extrato de código abaixo

```
public void start(Stage stage)
{
    TextField top = new TextField();
    TextField bottom = new TextField();

    // vincula a propriedade de texto do campo de texto inferior à propriedade do campo do topo
    // <instrução>

    VBox root = new VBox(10);
    root.getChildren().addAll(top, bottom);
    root.setAlignment(Pos.CENTER);

    Scene scene = new Scene(root, 480, 200);
    stage.setScene(scene); stage.setTitle("Exemplo de propriedade");
    stage.show();
}
```

Qual deverá ser a <instrução> à seguir aos comentários, que permite vincular as propriedades?

Resposta correta: `top.textProperty().bindBidirectional(bottom.textProperty());`

Considere o código na listagem seguinte:

```
public class ComboBoxExample extends Application
{
@Override
public void start(Stage stage)
{
final double WIDTH = 400; final double HEIGHT = 150;

  // <Instrução 1>
  box.getItems().addAll("Small", "Medium", "Large", "Extra large");
  box.setValue("Chooose your size");
  Label message = new Label();
  box.setOnAction(e -> message.setText("You have chosen: " +   // <Instrução 2>
                                       ));
  VBox root = new VBox(10); root.setPadding(new Insets(20, 20, 20, 20));
  root.setAlignment(Pos.TOP_CENTER);
  root.getChildren().addAll(box, message);
  Scene scene = new Scene(root, WIDTH, HEIGHT);
  stage.setScene(scene);
  stage.setTitle("Combo Box Example");
  stage.show();
}
```

Identifique nas propostas a seguir, quais delas devem substituir a `<Instrução 1>` e a `<Instrução 2>` na listagem.

A resposta correcta é: Instrução 1 → ComboBox<String> box = new ComboBox<>();

Instrução 2 → box.getValue()

Considere o código na listagem seguinte, que permite garantir que o conjunto de botões age como um grupo no qual apenas uma opção pode ser selecionada:

```
public void start(Stage stage) {
final double WIDTH = 350; final double HEIGHT = 200;

  RadioButton adultSingle = new RadioButton("Adult single");
  RadioButton childSingle = new RadioButton("Child single");
  RadioButton adultReturn = new RadioButton("Adult return");
  RadioButton childReturn = new RadioButton("Child return");

 _________ group = new __________();
  group.getToggles().addAll(adultSingle, childSingle, adultReturn, childReturn);
  Button submitButton = new Button("Choose your ticket");
  Label message = new Label();

  //...
}
```

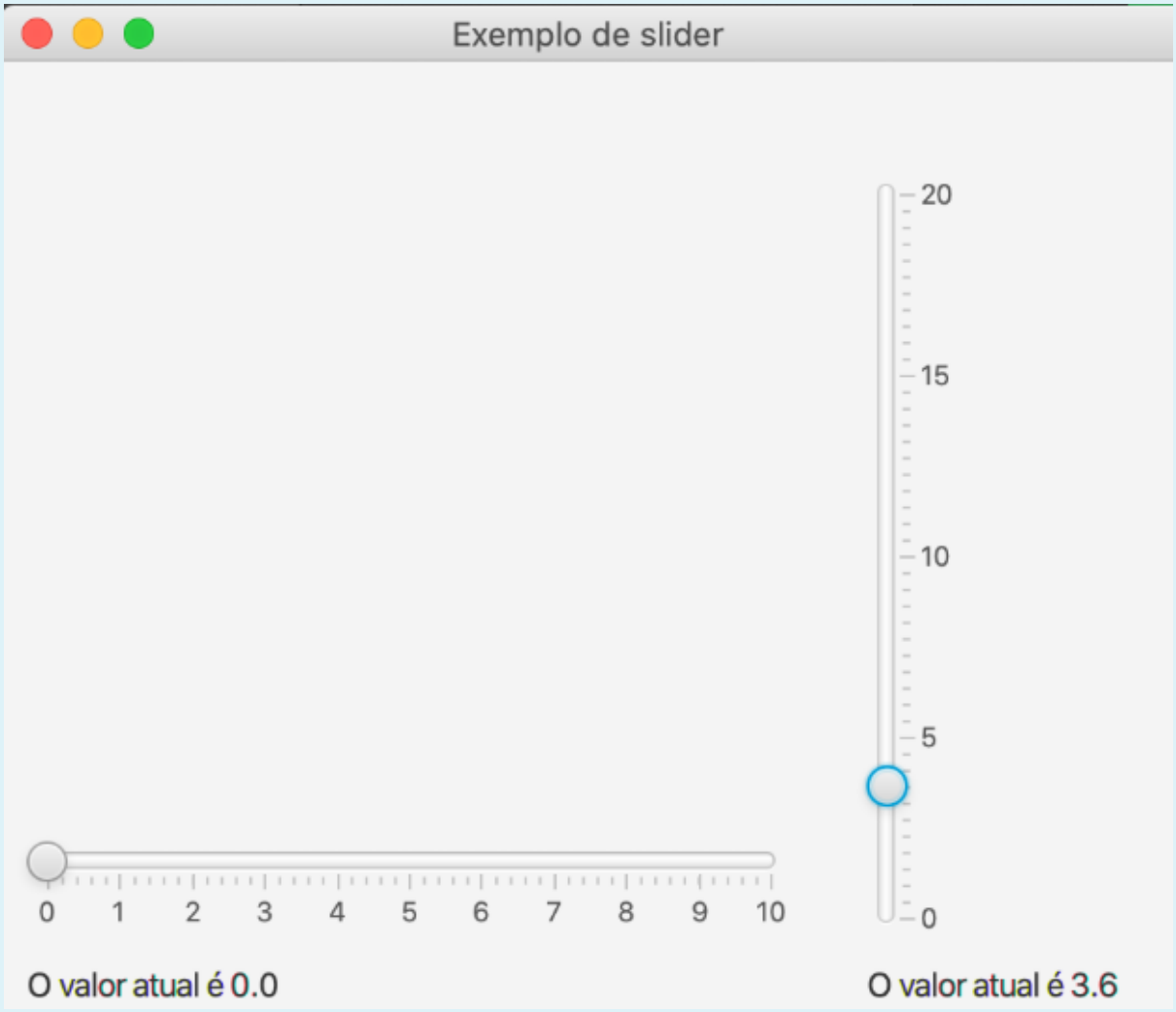
Indique que classe deve ser usada no lugar do símbolo ----------

Resposta correta: ToggleGroup

Para criar um menu, é necessário declarar obrigatoriamente objetos das seguintes classes

Respostas corretas: `MenuBar`, `Menu`, `MenuItem`

Considere o programa composto pela janela abaixo com dois controlos deslizantes `Slider`.



Considere o extrato de código a seguir que configura o controlo deslizante vertical:

```
Slider vertSlider = new Slider(0, 20, 0);
vertSlider.setMinHeight(vertSliderHeight);
vertSlider.setShowTickMarks(true);
vertSlider.setShowTickLabels(true);
vertSlider.setSnapToTicks(true);
vertSlider.setMajorTickUnit(5.0);
vertSlider.setMinorTickCount(10);
// <instrução>
```

Qual é a instrução em falta para finalizar a configuração das propriedades?

Resposta correta: `vertSlider.setOrientation(Orientation.VERTICAL);`

Se consideramos a classe `TextField` do JavaFX, os métodos `setText` e `getText` deste controlo permitem aceder

Resposta correta: as propriedades

Teste AC 9 – JavaFX – Janelas, Formas e Efeitos

Teste AC 10 – JavaFX - Animações