

Universidade Federal de Itajubá

1ª Prova de ECOP05 – Programação Aplicada

Nome:	Davi dos Santos Costa	Nº	2021010666
--------------	------------------------------	-----------	-------------------

Duração aproximada de 1:50 h. A interpretação faz parte da prova Data: 18/10/2021

Preencher as respostas na cor **vermelha**.

Salvar o arquivo em PDF com o nome: **prova1-NomeCompleto-NumeroMatricula.pdf**

1ª Questão (20 pts). Responda as questões a seguir:

- a) O que é, qual a principal aplicação e qual sintaxe utilizada para criar uma classe interna anônima em Java?

Classe anônima é a herança de determinada classe em um local exclusivo, ou seja, apenas naquele determinado ponto eu preciso redefinir minha classe-pai.

```
class Passaro{
Public void grunir(){...}
}
clas Gaivota{
    Passaro passaro = new Passaro(){
        Public void grunir(){...}
    };
}
```

- b) Qual sintaxe para utilizar um construtor de uma classe base dentro de uma classe derivada. Dê pelo menos um exemplo?

Usasse Extends para criação da classe derivada e super(); para se referir aos construtores da classe mãe

Exemplo :

Classe mãe:

```
public abstract class Figura {
Public int x0 , y0 ;
public Figura(int x, int y){ x0 = x; y0 = y; }
}
```

Classe derivada:

```
public class Retangulo extends Figura {
public Retangulo(int b, int a) { super(); largura = b; altura = a; }
}
```

- c) Explique as duas fases de compilação de programas Java.

A primeira é quando o código-fonte é traduzido em bytecodes ,onde tem uma portabilidade entre JVMs em diferentes plataformas do computador, e a segunda é durante a execução, quando os bytecodes são traduzidos em linguagem de máquina.

d) Como funciona o modelo de manipulação de eventos do Java?

Para manipular eventos vai precisar utilizar os listeners que são usado para “escutar” os eventos para poder tratar deles depois.

2ª Questão (5 pts). Explique para que serve cada uma das classes abaixo:

a) GridLayout

A classe Java GridLayout é usada para organizar os componentes em uma grade retangular. Um componente é exibido em cada retângulo.

b) ButtonGroup

É uma classe usada para gerenciar um conjunto de botões. Criar um conjunto de botões com o mesmo ButtonGroup objeto significa que "ligar" um desses botões desativa todos os outros botões do grupo.

c) BorderLayout

O BorderLayout é usado para organizar os componentes em cinco regiões: norte, sul, leste, oeste e centro. Cada região (área) pode conter apenas um componente.

d) JSlider

A classe Java JSlider é usada para criar o controle deslizante. Usando JSlider, um usuário pode selecionar um valor de um intervalo específico.

e) FlowLayout

Um layout de fluxo organiza os componentes em um fluxo direcional, muito parecido com as linhas de texto em um parágrafo. A direção do fluxo é determinada pela componentOrientation que define se vai ser da direita para a esquerda, ou da esquerda para a direita

3ª Questão (5 pts). Preencha as lacunas de cada uma das sentenças a seguir, utilizando as palavras abaixo:

atributos – main – classes – herança – encapsulamento

- a) Os objetos permitem a prática do **encapsulamento** — embora eles possam se comunicar entre si por meio de interfaces bem definidas, normalmente não têm autorização para descobrir como outros objetos são implementados.
- b) Os programadores Java concentram-se na criação de **classes**, que contêm campos e o conjunto de métodos que manipulam esses campos, além de fornecer serviços para clientes.
- c) Uma nova classe de objetos pode ser convenientemente criada por **herança** — a nova classe (chamada subclasse) começa com as características de uma classe existente (chamada superclasse), personalizando-as e talvez adicionando características próprias.
- d) O tamanho, forma, cor e peso de um objeto são considerados **atributos** da classe dele.
- e) Aplicativos Java iniciam a execução no método **main**.

4ª Questão (5 pts). – Para o programa listado a seguir, escrever o que será exibido como resultado da execução do mesmo. Escreva estes resultados na folha de questões:

```
1 // Questão 3a: Mystery.java
2 public class Mystery
3 {
4     public static void main( String[] args )
5     {
6         int i, j, k;
7
8         for ( i = 1; i <= 3; i++ )
9         {
10             for ( j = 1; j <= i+1 ; j++ )
11             {
12                 for ( k = 1; k <= i; k++ )
13                     System.out.print( '#' );
14                 System.out.println();
15             } // end inner for
16             System.out.println();
17         } // end outer for
18     } // end main
19 } // end class Mystery
```

Saida:

```
#
#

##
##
##

###
###
###
```

5ª Questão (5 pts). Preencha as lacunas em cada uma das seguintes afirmações:

- a) A classe **JMenu** é utilizada para criar um objeto de menu.
- b) O método **.addSeparator()** da classe JMenu coloca uma barra separadora em um menu.
- c) Os eventos JSlider são tratados pelo método **addChangeListener** da interface **JSliderChangeListener**.
- d) O método **setJMenuBar (bar);** anexa uma JMenuBar a um JFrame.

6ª Questão (10 pts). Elabore um aplicativo que lê uma linha de texto, tokeniza a linha com o método String **split** e imprime os tokens na ordem inversa. Utilize caracteres de espaço em branco como delimitadores.

```
Scanner entrada = new Scanner(System.in);
System.out.println("Digite um texto para ser dividido em seus tonkens
constituintes, e pressione enter:");
String texto = entrada.nextLine();
```

```
// Realizar o processamento da frase, separando seus tokens
```

```
String[] tokens = texto.split(" ");

//Mostrar os tokens na ordem inversa
System.out.println("Os tokens da frase são:\n");

for (int i = tokens.length ; i > 0 ; i--) {
    System.out.println(tokens[i-1]);
}
```

7ª Questão. (20 pts) Criar uma classe **JPoligonosRegulares** com pelo menos um construtor e os métodos **draw** e **fill**, que servirão para o objeto se criar, se desenhar e se preencher, respectivamente.

O construtor da classe deve receber os parâmetros necessários para caracterizar um polígono, (número de lados, tamanho e cores).

Os métodos **draw** e **fill** devem receber como parâmetro pelo menos o contexto gráfico para que o objeto possa ser desenhado ou preenchido. Ao ser chamado o método **fill** do objeto, ele deve ser apresentado preenchido com um gradiente de cores.

Utilizar a classe **GeneralPath** para a geração do polígono a ser desenhado/preenchido.

(10 pts) Utilize esta classe em um programa que irá criar **n** objetos do tipo **JPoligonosRegulares** com número de lados, tamanho e cores aleatórios e desenhar esses objetos em posições aleatórias em uma janela. O programa deve possuir uma caixa de texto (**JTextField**) para permitir que o usuário digite o número de polígonos a ser desenhado. Crie um **Timer** para permitir que a tela fique sendo atualizada a cada 3 segundos.

8ª Questão (20 pts). – Para cada uma das afirmações a seguir, classificá-las como verdadeira ou falsa, caso a afirmação seja falsa, destacar o que a torna falsa.

- (F) O método **drawArc** da classe **Graphics** não aceita valores negativos para o parâmetro que especifica o ângulo.
- (V) Construtores de uma classe base são herdados nas classes derivadas.
- (V) Uma classe de adaptadores implementa uma interface, fornecendo uma implementação padrão (com o corpo de um método vazio) de todos os seus métodos.
- (V) Um mesmo *array* não pode armazenar dados de tipos diferentes em seu interior.
- (V) Se apenas um elemento de um *array* de *int*, for passado como parâmetro para um método que vai modificar esse valor em seu interior, após completar a execução do método o *array* irá conter o valor da variável modificada.
- (V) Em Java, o separador padrão de argumentos na linha de comando é a espaço.
- (V) Os valores de ponto flutuante que aparecem no código-fonte são conhecidos como literais de ponto flutuante e são tipos *double* por padrão.
- (V) O relacionamento “É-Um” é implementado utilizando herança.
- (F) Os dois primeiros parâmetros do método **drawOval** da classe **Graphics** especificam as coordenadas do canto superior esquerdo da Oval.

- (F) Em Java, por padrão, a coordenada do pixel (0, 0) está localizada no centro do monitor.
- (F) Em uma declaração de array, especificar o número de elementos entre os colchetes da declaração (por exemplo, `int[12] c;`) é uma operação válida em Java.
- (F) O método `Graphics fillArc` utiliza graus para especificar o ângulo.
- (V) O layout padrão de um objeto da classe `javax.swing.JPanel` é o `java.awt.FlowLayout`.
- (V) Quando uma classe implementa uma interface, a classe possuirá o relacionamento “é um” com essa interface. Todas as subclasses diretas e indiretas dessa classe herdam essa interface. Portanto, um objeto de uma classe que estende uma classe de adaptadores de evento é um objeto do tipo ouvinte de eventos correspondente, por exemplo, um objeto de uma subclasse de `MouseAdapter` é um `MouseListener`.
- (V) Parênteses vazios após um nome de método indicam que o método não requer nenhum parâmetro para executar sua tarefa.
- (V) Um `BorderLayout` limita um Container a conter no máximo cinco componentes — um em cada região.
- (V) Toda declaração de classe que começa com o modificador de acesso `public` deve ser armazenada em um arquivo que tenha o mesmo nome da classe com a extensão de nome de arquivo `.java`.
- (V) No sistema de coordenadas do Java, as coordenadas x aumentam da esquerda para a direita e as coordenadas y, de baixo para cima.
- (F) Quando o programador cria um `JFrame`, no mínimo um menu deve ser criado e adicionado ao `JFrame`.