

Algoritmos e Programação de Computadores II - COM120 - Turma 003

Página Inicial

Avísos

Cronograma

Atividades

Fóruns

Collaborate

Calendário Lives

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações Gerais para a Avaliação

Exame

Documentos e informações gerais

Gabaritos

Facilitadores da disciplina

Repositório de REAs

Página Inicial1

Revisar envio do teste: Semana 7 - Atividade Avaliativa

Usuário

LIZIS BIANCA DA SILVA SANTOS

Curso

Algoritmos e Programação de Computadores II - COM120 - Turma 003

Teste

Semana 7 - Atividade Avaliativa

Iniciado

01/06/23 19:22

Enviado

01/06/23 19:33

Data de vencimento

02/06/23 05:00

Status

Completada

Resultado da tentativa

10 em 10 pontos

Tempo decorrido

10 minutos

Resultados exibidos

Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 1

1,66 em 1,66 pontos

Uma Interface Gráfica do Usuário (GUI) trata-se de um ambiente gráfico cujo propósito é simplificar o uso do computador por meio de representações gráficas e intuitivas do sistema operacional. É um ambiente composto por diversos aspectos que propiciam ações pelo uso de atalhos do teclado, *mouse* e também pelo toque.

Análise as alternativas a seguir e marque a que demonstra os aspectos que formam tal ambiente gráfico.

Resposta Selecionada:

e

 Janelas, ícones, *menus* e ponteiros.

Respostas:

a

Janelas e entradas USB.

b

Ícones, *menus*, janelas e teclas.

c

Entradas USB, janelas e ícones.

d

Botões, *drives* e ponteiros.

e

Janelas, ícones, *menus* e ponteiros.

Comentário da resposta:

JUSTIFICATIVA
Uma GUI forma-se a partir de janelas, ícones, *menus* e ponteiros, que podem ser acionados por meio do *mouse*, do toque ou de atalhos no teclado, tudo é pensado de forma a conduzir a interação do usuário final com os dispositivos digitais com a utilização de elementos gráficos; eles são definidos como a "cara" do programa.

Pergunta 2

1,66 em 1,66 pontos

O gerenciador de *layout pack* vai inserindo os *widgets* na ordem em que forem empacotados, fazendo a chamada do método pack() de cima para baixo, contudo existem diversas opções para empacotar o componentes, como preencher, expandir e lado, para controlar esse gerenciador de geometria.

Considerando as opções para empacotar os componentes, avalie as afirmações a seguir, em relação às suas funções, e as relacione adequadamente aos termos a que se referem.

1. *Side*.

2. *Fill*.

3. *Expand*.

I. Indica o lado em que se deseja adicionar o *widget*.

II. Preenche um espaço no *widget*.

III. Define se o *widget* vai preencher todo o espaço extra do *container* ou não.

Assinale a alternativa que relaciona adequadamente os dois grupos de informações.

Resposta Selecionada:

c

 1-I; 2-II; 3-III.

Respostas:

a

1-I; 2-III; 3-II.

b

1-III; 2-I; 3-II.

c

1-I; 2-II; 3-III.

d

1-II; 2-I; 3-III.

e

1-III; 2-II; 3-I.

Comentário da resposta:

JUSTIFICATIVA
A sentença I enquadra-se no conceito 1, porque o *side* é a opção para empacotar componentes que serve para indicar a forma de como ele será adicionado (TOP, BOTTOM, LEFT e RIGHT), isto é, os lados para adicionar o *widget*. Se quer um alinhamento horizontal, use *side*=LEFT, por exemplo. A sentença II enquadra-se no conceito 2, porque o *fill* é a opção para empacotar componentes cuja finalidade é a de preencher um espaço, e é possível utilizar como argumento X, Y ou BOTH, para cobrir todo o espaçamento horizontal (X), vertical (Y) ou ambos (BOTH). A sentença III enquadra-se no conceito 3, porque o *expand* é a opção para empacotar componentes que define se o *widget* vai preencher todo o espaço extra do *container* ou não, pode passar como YES ou NO.

Pergunta 3

1,66 em 1,66 pontos

As ações executadas em resposta a um evento são os *event handlers* e, ao estruturar uma interface gráfica em classes, é possível escrevê-los no formato de métodos, que são as funções definidas dentro da classe para depois atribuir o evento ao *event handler*.

Após análise do trecho apresentado, avalie as asserções a seguir e a relação proposta entre elas.

I. Os *event handlers* são as ações executadas em resposta a um evento, ao associar <Button-1> a um *event handler*, não é preciso soltar o botão do *mouse* para que ele seja executado.

PORQUE
II. O *event handler* é chamado por meio da ação de pressionar o botão do *mouse*, pois os eventos clicar e soltar do *mouse*, em um *widget*, agem de forma diferente em Tkinter.

A respeito dessas asserções, assinale a alternativa correta.

Resposta Selecionada:

c

 As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

Respostas:

a

A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

b

As asserções I e II são falsas.

c

As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

d

As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.

e

A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.

Comentário da resposta:

JUSTIFICATIVA
A asserção I é verdadeira, pois o *event handler* é a ação a ser executada quando um evento deve ser disparado, sendo geralmente um método ou uma função com uma assinatura específica chamada pelo evento. A asserção II é verdadeira, visto que, em interfaces gráficas, o evento mais comum é o clique do *mouse*, por esse motivo, deve existir uma maneira rápida que associe esse evento frequente a um *event handler*, não exigindo a aplicação do método *bind*. A asserção II justifica a primeira, pois, sendo o *event handler* um manipulador de evento, ele é que responde quando um evento é disparado, manipulando o evento e provocando alguma ação.

Pergunta 4

1,66 em 1,66 pontos

O método *mainloop* bloqueia a execução da janela atual usando um *event loop* do *Tkinter*, ou seja, trava a *thread* impedindo que o código abaixo da sua chamada seja executado, até que a janela seja fechada. Com isso, o *Tkinter* consegue disparar respostas a eventos na instância atual da janela que chamou o *mainloop* (*keypress* ou *click*), não permitindo a execução de códigos posteriores até que a janela seja fechada.

1 from Tkinter import *

Fonte: Elaborada pela autora.

2 class Janela:

Fonte: Elaborada pela autora.

3 def __init__(self, instancia_de_Tk):

Fonte: Elaborada pela autora.

4 pass

Fonte: Elaborada pela autora.

5 raiz=Tk()

Fonte: Elaborada pela autora.

6 Janela(raiz)

Fonte: Elaborada pela autora.

7 raiz.mainloop()

Fonte: Elaborada pela autora.

Sendo assim, analise as afirmativas a seguir.

I. Na linha 7, o método *mainloop* faz com que a interface fique aberta aguardando pelo acontecimento de eventos.

II. Ao aplicar o método *mainloop*, a janela permanece aberta até ocorrer algum evento capaz de desmanchá-la.

III. Nesse exemplo, não são criados automaticamente botões de minimizar, maximizar e de fechar.

IV. Não há necessidade de chamar a função *mainloop* para que a aplicação entre no modo de tratamento de eventos.

Está correto que se afirma em:

Resposta Selecionada:

e

 I e II, apenas.

Respostas:

a

I, II e IV, apenas.

b

II e IV, apenas.

c

III e IV, apenas.

d

I, II e III, apenas.

e

I e II, apenas.

Comentário da resposta:

JUSTIFICATIVA
A afirmativa I é correta, porque, na linha 7, o método *mainloop* faz com que a interface fique aberta, esperando acontecer algum evento; e, caso esse evento venha a ocorrer e seja apenas um clique em alguma área fora da interface, nada acontecerá, porém se algum botão for pressionado, o *event handler* será acionado. A afirmativa II é correta, pois, aplicando o método *mainloop*, a janela fica aberta até que aconteça um clique no botão de fechar. A afirmativa III é incorreta, já que, nesse trecho de código, os botões de minimizar, maximizar e fechar são criados automaticamente na GUI devido a se estar trabalhando com Tkinter. A afirmativa IV é incorreta, pois a função *mainloop* deve ser requisitada para que a aplicação entre no modo de tratamento de eventos.

Pergunta 5

1,68 em 1,68 pontos

A interação entre código e usuário pode ser realizada por meio de cliques em botões e por meio de áreas em forma de retângulo, em que é possível a escrita de algum texto, portanto uma forma de capturar dados do usuário é por meio do _____ em que os dados informados são captados em forma de _____, e o que permite trabalhar tais dados dentro código é o método _____. fornecido pelo Entry que aceita a maior parte das configurações para botões e *labels*.

Preencha as lacunas escolhendo a alternativa correta.

Resposta Selecionada:

a

widget text — *string* — *get*.

Respostas:

a

widget text — *string* — *get*.

b

widget entry — *string* — *get*.

c

widget entry — *char* — *delete*.

d

widget entry — *string* — *insert*.

e

widget frame — *string* — *bind*.

Comentário da resposta:

JUSTIFICATIVA
A primeira lacuna é completada pelo termo "widget entry" usado para inserir ou exibir uma única linha de texto e que permite que os dados informados em uma tela sejam captados em forma de *string*. A segunda lacuna é completada pelo termo "string", pois é dessa forma que as informações inseridas nas áreas de preenchimento de dados dentro do código do método *mainloop*, a janela fica aberta até que aconteça um clique no botão de fechar. A afirmativa III é incorreta, já que, nesse trecho de código, os botões de minimizar, maximizar e fechar são criados automaticamente na GUI devido a se estar trabalhando com Tkinter. A afirmativa IV é incorreta, pois a função *mainloop* deve ser requisitada para que a aplicação entre no modo de tratamento de eventos.

Pergunta 6

1,68 em 1,68 pontos

O *widget Frame* pode ser entendido como um *container* retangular com a principal finalidade de servir de base para a instalação de outros *widgets*, cujo posicionamento depende do fornecimento de informações precisas em relação às coordenadas x e y para o canto superior esquerdo do *widget*.

Após análise do conceito sobre o *widget Frame*, avalie as asserções a seguir e a relação proposta entre elas.

I. Ao utilizar uma função up() para mover a coordenada y em 10 unidades, o que acontece é o que o valor de y diminui 10 unidades.

PORQUE
II. O sistema de coordenadas na tela de desenho não é o mesmo adotado pelo sistema de coordenadas.

A respeito dessas asserções, assinale a alternativa correta.

Resposta Selecionada:

e

 As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

Respostas:

a

A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.

b

As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.

c

As asserções I e II são falsas.

d

A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.

e

As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

Comentário da resposta:

JUSTIFICATIVA
A asserção I é verdadeira, pois a função up() move a posição da coordenada (x,y) para cima. Portanto o que acontece na tela de desenho para mover a coordenada y em, por exemplo, 10 ou 20 unidades, é que y será diminuído em 10 ou 20 unidades. Esse fato é diferente considerando um sistema de coordenadas típico, o que indica que y aumentaria em 10 ou 20 unidades. A asserção II é verdadeira, pois a posição nas coordenadas (0,0) encontra-se no canto superior esquerdo da tela de desenho. As coordenadas x aumentam para a direita e as coordenadas y aumentam para baixo na tela de desenho. Por isso, mover para cima significa diminuir a coordenada y, que é o que faz a função up(). O sistema de coordenadas na tela de desenho distingue-se do adotado pelo sistema de coordenadas, como exemplificado na justificativa da asserção I. A asserção II justifica a primeira, pois a função up() para as coordenadas na tela de desenho funciona movimentando o objeto sem desenhá-lo, com o aumento da coordenada x para a direita e o aumento da coordenada para baixo.