

Algoritmos e Programação de Computadores I - COM110 - Turma 003

Página Inicial

Avisos

Cronograma

Atividades

Fóruns

Colaborate

Calendário Lives

Notas

Menu das Semanas

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

Orientações para realização da prova

Documentos e Informações gerais

Gabaritos

Referências da disciplina

Facilitadores da disciplina


Repositório de REA's

Página Inicial1

Revisar envio do teste: Semana 3 - Atividade Avaliativa

Usuário	LIZIS BIANCA DA SILVA SANTOS
Curso	Algoritmos e Programação de Computadores I - COM110 - Turma 003
Teste	Semana 3 - Atividade Avaliativa
Iniciado	13/03/23 19:42
Enviado	13/03/23 19:49
Data de vencimento	17/03/23 05:00
Status	Completa
Resultado da tentativa	8,58 em 10 pontos
Tempo decorrido	7 minutos
Instruções	Olá, estudante! 1. Para responder a esta atividade, selecione a(s) alternativa(s) que você considerar correta(s); 2. Após selecionar a resposta correta em todas as questões, vá até o fim da página e pressione "Enviar teste". 3. A cada tentativa, você receberá um novo conjunto de questões diferentes para que você responda e tente alcançar melhores resultados. Pronto! Sua atividade já está registrada no AVA.
Resultados exibidos	Todas as respostas, Respostas enviadas, Respostas corretas, Comentários, Perguntas respondidas incorretamente

Pergunta 11,42 em 1,42 pontos

 Na linguagem Python, é possível trabalhar com diversos tipos de dados, como números inteiros, reais, valores lógicos booleanos, sequências de caracteres, listas, dentre outros.

Com base nesse tema, avalie as expressões a seguir.

```
>>> x = 5
>>>y = 3,2
>>>nome = 'João'
>>>nomes = [nome, 'Maria', 'Felipe', 'José']
```

Com relação aos tipos de dados utilizados, avalie as afirmativas a seguir.

- I. O comando `type(x)` irá retornar `<class 'int'>`.
II. O comando `type(y)` irá retornar `<class 'int'>`.
III. O comando `type(nome)` irá retornar `<class 'str'>`.
IV. O comando `type(nomes)` irá retornar `<class 'list'>`


Está correto o que se afirma em:

Resposta Selecionada: ☒ a. I, III e IV, apenas.

Respostas: ☒ a. I, III e IV, apenas.
☐ b. I e II, apenas.
☐ c. I, II e III, apenas.
☐ d. II, apenas.
☐ e. I, II, III e IV.

Comentário da resposta: **JUSTIFICATIVA**
A afirmativa I está correta, pois o comando `type(x)` é responsável por analisar a variável passada como parâmetro e retornar seu tipo de dados e, nesse caso, retornará `<class 'int'>`. A afirmativa II está incorreta, pois o resultado do comando `type(y)`, que é responsável por analisar a variável passada como parâmetro e retornar seu tipo de dados, nesse caso, retornará será `<class 'float'>`. A afirmativa III está correta, pois o resultado do comando `type(nome)`, que analisa os tipos de dados da variável passada como parâmetro, será `<class 'str'>`. A afirmativa IV está correta, pois o resultado do comando `type(nomes)`, que analisa os tipos de dados da variável passada como parâmetro, será `<class 'list'>`.

Pergunta 20 em 1,42 pontos

 As bibliotecas contendo funcionalidades extras estão presentes em grande parte das linguagens de programação e, na Python, especificamente, existe um conjunto de módulos presente em sua biblioteca padrão que traz grande poder para a linguagem.

Considerando a biblioteca padrão Python, identifique se são (V) verdadeiras ou (F) falsas as afirmativas a seguir.

- I. () Os módulos da biblioteca padrão já estão disponíveis na linguagem Python, bastando apenas acioná-los pelo nome do método desejado.
II. () Para usar os módulos da biblioteca padrão Python, é necessário usar o comando `"import"` seguido do nome do módulo desejado.
III. () Para usar uma função de um determinado módulo, é necessário especificar o módulo e a função, em conjunto, separados por um ponto.
IV. () A função `math.sqrt()` pertence ao módulo *math* que, por sua vez, pertence à biblioteca padrão Python.


Assinale a alternativa que apresenta a sequência correta.

Resposta Selecionada: ☒ a. V, F, V, V.

Respostas: ☐ a. V, F, V, V.
☒ b. F, V, V, V.
☐ c. F, F, F, V.
☐ d. V, F, V, F.
☐ e. V, V, F, F.

Comentário da resposta: **JUSTIFICATIVA**
A afirmativa I é falsa, antes de chamar um método pertencente a um módulo da biblioteca padrão Python, é necessário importar o módulo com o comando `"import"`. A afirmativa II é verdadeira, pois o compilador irá verificar se o método chamado faz parte da biblioteca padrão da linguagem ou de outra biblioteca que precisará ser importada com a palavra reservada `"import"`. A afirmativa III é verdadeira, pois a sintaxe correta para a chamada de uma função em linguagem Python necessita da especificação do módulo antes do nome da função, e eles devem ser separados por um ponto. A afirmativa IV é verdadeira, pois a função `math.sqrt()`, utilizada para calcular a raiz quadrada de um número, faz parte do módulo *math* e da biblioteca padrão Python.

Pergunta 31,42 em 1,42 pontos

 Atualmente, uma importante área do conhecimento humano é a ciência de dados, que tem como apoio ferramentas como a linguagem Python. As listas em Python têm um conjunto de métodos manipuladores que auxilia na inserção, recuperação, alteração e remoção de elementos.

Considere a seguinte lista:

```
>>> nomes = ['João', 'Maria', 'José', 'Marta']
```

Com base na lista apresentada, assinale a alternativa correta.

Resposta Selecionada: ☒ c. O método `nomes.append('Priscila')` irá inserir um novo elemento no final da lista.

Respostas: ☐ a. Os métodos *insert* e *append* têm a mesma funcionalidade para as listas.
☐ b. O método `pop()` irá ordenar a lista.
☒ c. O método `nomes.append('Priscila')` irá inserir um novo elemento no final da lista.
☐ d. O método `nomes.count('João')` retornará o valor 0.
☐ e. O método `nomes.append('Priscila')` irá inserir o elemento no início da lista.

Comentário da resposta: **JUSTIFICATIVA**
O método `count('João')` verifica a quantidade de elementos 'João' que existem na lista e, nesse caso, é 1. O método `nomes.append('Priscila')` irá inserir o elemento no final da lista, e não no início. Os métodos *insert* e *append* são diferentes, pois o método *insert* permite escolher em que ponto da lista deseja-se inserir o elemento. O método `pop()` remove o último elemento da lista, e não realiza ordenação. A ordenação é realizada pelo método `sort()`.

Pergunta 41,42 em 1,42 pontos

 O uso de estruturas de dados, como listas e tuplas, é de grande utilidade para as linguagens de programação, especialmente para a Python, que explora esses recursos com grande eficiência.

Com base nas listas e tuplas, selecione a alternativa correta:

Resposta Selecionada: ☒ b. As tuplas são iguais às listas, exceto pela característica de serem imutáveis, não permitindo a alteração dos seus elementos.

Respostas: ☐ a. As listas em Python são sempre imutáveis, não permitindo a alteração dos seus elementos.
☒ b. As tuplas são iguais às listas, exceto pela característica de serem imutáveis, não permitindo a alteração dos seus elementos.
☐ c. As tuplas em Python são sempre mutáveis, permitindo a alteração dos seus elementos.
☐ d. As tuplas podem ser criadas em Python com o uso de colchetes separando os objetos por vírgulas.
☐ e. As listas em Python são recursos para tratamento específico de dados numéricos.

Comentário da resposta: **JUSTIFICATIVA**
As listas em Python permitem a criação de conjuntos de qualquer tipo de dados, inclusive numéricos. Uma das principais características das listas é que elas são mutáveis e permitem a alteração de seus elementos. As tuplas, ao contrário das listas, são imutáveis e não permitem a alteração de seus elementos. A criação de tuplas em Python ocorre com o uso de parênteses separando os objetos por vírgulas.

Pergunta 51,42 em 1,42 pontos

 A linguagem Python tornou-se popular pela sua capacidade de tratar dados de forma transparente e legível. O uso de listas em Python permite a aplicação de diversos operadores que são comuns ao tipo de dados *string*.

Considere a lista a seguir:

```
>>> nomes = ['João', 'Maria', 'José', 'Marta']
```


Com base na lista apresentada, selecione a alternativa correta.

Resposta Selecionada: ☒ b. O comando `nomes[-1]` retorna o elemento 'Marta'.

Respostas: ☐ a. O comando `nomes[4]` retorna o elemento 'Marta'.
☒ b. O comando `nomes[-1]` retorna o elemento 'Marta'.
☐ c. O comando `nomes[-2]` retorna o elemento 'Maria'.
☐ d. O comando `nomes[-1]` não retorna algum elemento.
☐ e. O comando `nomes[1]` retorna o elemento 'João'.

Comentário da resposta: **JUSTIFICATIVA**
Em Python, os elementos das listas podem ser recuperados pelo seu índice, lembrando que o primeiro elemento da lista tem o índice 0. Nesse caso, o primeiro índice da lista, `nomes[0]`, retornará 'João'. Os índices negativos levam a lista para um processo cíclico de traz para a frente o comando `nomes[-1]`, que levará ao último elemento, que é 'Marta'. O comando `nomes[4]` retornará um erro, pois não existe o elemento de índice 4 na lista, considerando que o primeiro elemento da lista é o índice 0. O comando `nomes[-2]` retornará 'José'.

Pergunta 61,45 em 1,45 pontos

 Considerando as aplicações abrangentes da linguagem, surge a biblioteca padrão Python com o intuito de trazer aplicabilidade e produtividade para o desenvolvimento de *software* com métodos de uso comuns.

Com base na biblioteca padrão Python, analise as asserções a seguir e as relações propostas entre elas.

I. A biblioteca padrão Python tem um vasto conjunto de funcionalidades que estão organizadas em mais de 200 módulos.

PORQUE

II. Cada módulo contém funções e classes específicas para um domínio de aplicação e essa organização permite maior produtividade para os desenvolvedores no uso da biblioteca.


Analisando as asserções anteriores, conclui-se que:

Resposta Selecionada: ☒ b. as duas asserções são verdadeiras, e a segunda justifica a primeira.

Respostas: ☐ a. a primeira asserção é verdadeira, e a segunda é falsa.
☒ b. as duas asserções são verdadeiras, e a segunda justifica a primeira.
☐ c. a primeira asserção é falsa, e a segunda é verdadeira.
☐ d. as duas asserções são verdadeiras, e a segunda não justifica a primeira.
☐ e. as duas asserções são falsas.

Comentário da resposta: **JUSTIFICATIVA**
A primeira asserção é verdadeira, pois a linguagem Python incorporou importantes recursos de aplicações computacionais de diversas áreas possuindo mais de 200 módulos na biblioteca padrão sendo essa a forma de organização. A segunda asserção é verdadeira e justifica a primeira, pois cada um dos módulos tem funções específicas que são classificadas conforme seu domínio de aplicação. A segunda proposição justifica a primeira tendo em vista que a organização em módulos é um fator de viabilização da biblioteca em termos de usabilidade por parte dos desenvolvedores, caso contrário, a pesquisa de funções em todo o conjunto da biblioteca padrão Python seria muito improdutiva.

Pergunta 71,45 em 1,45 pontos

 A definição dos tipos de dados de forma otimizada garante mais efetividade das soluções frente às limitações dos sistemas computacionais, por isso as linguagens de programação, em especial, a Python, utilizam tipos especiais de dados.

Com base no uso do tipo *fractions*, analise as asserções a seguir e as relações propostas entre elas.

I. Uma característica importante do tipo *fraction* é sua capacidade de representar números muito maiores se compararmos com a capacidade do tipo *float*.

PORQUE

II. O uso do tipo de dados *float* é uma opção para as expressões que exigem maior velocidade, em vez de uma capacidade maior de representação do número.

Analisando as asserções anteriores, conclui-se que:

Resposta Selecionada: ☒ e. as duas asserções são verdadeiras, e a segunda não justifica a primeira.

Respostas: ☐ a. as duas asserções são verdadeiras, e a segunda justifica a primeira.
☐ b. a primeira asserção é verdadeira, e a segunda é falsa.
☐ c. a primeira asserção é falsa, e a segunda é verdadeira.
☐ d. as duas asserções são falsas.
☒ e. as duas asserções são verdadeiras, e a segunda não justifica a primeira.

Comentário da resposta: **JUSTIFICATIVA**
A primeira asserção é verdadeira, pois existem situações em que o tipo de dados *float* não é suficiente para representar um determinado número, por isso foi criado o módulo *fractions* e o respectivo tipo *fraction*. A segunda asserção é verdadeira e não justifica a primeira, pois, apesar da menor representatividade em relação à extensão numérica, o tipo *float* possibilita maior velocidade de implementação do que o tipo *fraction*. As asserções apenas explicam dois tipos de dados similares, destacando suas principais diferenças.