Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 1: Tema - 1 . PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO: MOTIVAÇÃO E PRELIMINARES

3 Objetivos

Entender as motivações atuais para estudar linguagens de programação.

4 Tópicos

1 .1 RAZÕES PARA ESTUDAR CONCEITOS DE LINGUAGENS DE PROGRAMAÇÃO

5 Procedimentos de ensino-aprendizagem

Apontar as motivações atuais para estudar linguagens de programação. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Suponha que um grupo de estudantes de diversas áreas como engenharia, computação e saúde estejam reunidos e discutindo sobre a importância da programação de computadores para cada uma de suas áreas. Como os estudantes de cada área iria descrever essa importância?

Metodologia:
Iniciar exibindo o vídeo "Por que todos deveriam aprender a programar?" [2]. O professor deve destacar que todas as pessoas podem se beneficiar ao aprender a programar apresentando benefícios nas áreas econômicas, melhoria na qualidade de vida, automatização de tarefas cotidianas, utilização mais eficiente do tempo para outras tarefas que demandam ações que só podem ser realizadas por pessoas entre outras.

Atividade verificadora de aprendizagem:
O professor deve dividir a turma em grupos temáticos (engenharia, economia, saúde, ciências sociais, computação, por exemplo) e cada grupo deve, após um pequeno tempo para conversar, apresentar aos outros alunos suas conclusões sobre a importância da programação para cada uma das diversas áreas. O professor deve apenas mediar as sugestões dos alunos, e se ater a guiá-los no processo de construção coletiva da solução para situação-problema. O feedback deverá ser realizado pelo professor durante a aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 1 (Preliminares). Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Vídeo "Por que todos deveriam aprender a programar?". Disponível em: https://www.youtube.com/watch?v=mHW1Hsqlp6A

8 Aprenda +

[3] Artigo: "9 razões para aprender programação". Disponível em: https://www.digitalhouse.com/br/blog/9-motivos-aprender-programar-programador.

Atividade Autônoma Aura:

Questão 1. Aprender linguagens de programação se tornou indispensável na nossa sociedade. Segundo reportagem publicada no Olhar Digital: "A necessidade de um segundo idioma é praticamente indispensável para que o profissional se mantenha competitivo no atual mercado de trabalho. Seja qual for a área de atuação. E isso não é de hoje. Se você não fala inglês ou espanhol, pode ter certeza: seu currículo vai ficar ali, separado em um segundo bloco. A novidade é que está chegando a hora de se preparar para aprender mais uma linguagem: programação, é o idioma da inovação. E promete se tornar habilidade básica do profissional do futuro. Ou até já do presente?" Disponível em: https://olhardigital.com.br/2020/05/23/videos/programacao-passa-a-ser-diferencial-em-multiplas-areas-do-mercado-de-trabalho-2/. Acessado em: 19/01/2021.

Nesse sentindo, qual(s) das vantagens abaixo podem ser relacionadas a habilidades que podem ser adquiridas ao aprender várias linguagens de programação?

I - Aumento da capacidade de expressar ideias

II - Melhor entendimento da importância da implementação

III - Ser especialista em Inteligência Artificial

a) I e III

b) II e III

c) I e II

d) Apenas I

e) Todas

Questão 2. Existem muitas linguagens de programação disponíveis no mercado com as mais diversas características e aplicações. No entanto, as linguagens de programação, possuem restrições de tipos de

estrutura de controle, estrutura de dados e abstrações que podem ser utilizadas. Nesse contexto, análise as afirmações a seguir:

I - Ao aprender várias linguagens de programação, é possível que um problema possa ser resolvido mais facilmente devido a adequação a uma linguagem específica.

II - Não há necessidade de se aprender mais do que uma linguagem de programação haja vista que as estruturas possuem equivalentes em todas elas

III - Aprender diversas linguagem de programação propicia um melhor embasamento para decidir qual deve ser utilizada para resolver um determinado problema.

É (são) verdadeira(s):

a) I e III

b) I e II

c) II e III

d) Todas são verdadeiras

e) Nenhuma afirmação é verdadeira

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 2: Tema - 1 . PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO: MOTIVAÇÃO E PRELIMINARES

3 Objetivos

Discutir as principais áreas de aplicação da computação e as linguagens associadas.

4 Tópicos

1.1 RAZÕES PARA ESTUDAR CONCEITOS DE LINGUAGENS DE PROGRAMAÇÃO
1.2 DOMÍNIOS DE PROGRAMAÇÃO

5 Procedimentos de ensino-aprendizagem

Explicar as razões para se estudar conceitos de linguagens de programação, os domínios de programação, os critérios de avaliação das linguagens de programação, e fatores que influenciam o projeto de linguagens. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Suponha que um programador necessita decidir qual linguagem de programação deve utilizar para desenvolver um App simples para um celular Android, que possui apenas 3 telas que deveriam deslizar para os lados. Esse App não possui cadastro, objetos em 3D, interações complexas nem cálculos matemáticos que precisem ser implementados. Esse programador implementou esse App utilizando a linguagem C# usando a Unity 3D, que é uma game engine, que possui motor gráfico embutido, que nesse caso não é utilizado. Isso faria com que o App tivesse 50 MB no lugar dos 2MB que deveria ter. Quais foram os problemas que levaram a essa escolha equivocada da linguagem?

Metodologia:
O professor poderá utilizar ferramentas de retorno imediato como o Mentimeter disponível em https://www.mentimeter.com. Nesse caso, o professor solicita aos alunos que escrevam os problemas e após isso ele irá discutir com a turma quais seriam as questões de projeto que deveriam ser levadas em consideração na escolha da linguagem para o desenvolvimento.

Atividade verificadora de aprendizagem:
Recomenda-se os exercícios
propostos nas páginas 30, 31 e 32 do livro Conceitos de Linguagens de Programação [1]. A atividade deve ser corrigida pelo professor durante a aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/ , páginas 30, 31 e 32.

8 Aprenda +

[2] Vídeo "O poder do paradigma", disponível em https://www.youtube.com/watch?v=Qba6MgpXJfY. (Ative a legenda e a tradução automática para português).

[3] Vídeo "Programação através de Paradigmas", disponível em https://www.youtube.com/watch?v=Pg3UeB-5FdA (Ative a legenda e a tradução automática para português).

Atividade Autônoma Aura:

Questão 1: A adequação de linguagens de computadores a aplicações específicas é uma prática importante devido aos objetivos diversos com que as linguagens de programação são desenvolvidas. Diante dessa situação, relacione as colunas:

1. Aplicações Científicas

2. Aplicações Empresarias

3. Inteligência Artificial

4. Software para Web

( ) Linguagem LISP surgida em no final da década de 1950, Python

( ) Linguagem COBOL

( ) Linguagem de programação FORTRAN

( ) PHP e HTML

Qual a sequência de relacionamento correto?

a) 1 -3 -2 -4

b) 3- 1 - 2 - 4

c) 2 - 3 - 1 - 4

d) 3 - 2 - 1 - 4

e) 4 -2 - 1 - 3

Questão 2: Os cientistas de dados são uma nova geração de especialistas analíticos que possuem as habilidades técnicas para resolver problemas complexos "e a curiosidade para explorar quais problemas precisam ser resolvidos" Cientistas de Dados: Quem são, o que fazem e por que você quer ser um. Disponível em: https://www.sas.com/pt_br/insights/analytics/cientistas-de-dados.html#:~:text=Os%20cientistas%20de%20dados%20s%C3%A3o%20uma%20nova%20gera%C3%A7%C3%A3o,cientistas%20da%20computa%C3%A7%C3%A3o%20e%20parte%20observadores%20de%20tend%C3%AAncias. Acessado em 20/01/2021.

Nesse contexto, sabemos que os cientistas de dados lidam com grande quantidade de dados que precisam trabalhados para gerar insights sobre eles. Dada as especificidades, linguagens de programação específicas são utilizadas para processamento desses dados. Assinale qual alternativa contém linguagens que são primordialmente utilizadas para essa função ou que estão tendo grande crescimento na sua utilização.

a) FORTRAN e C++

b) C e Assembly

c) PHP e Java Script

d) LISP e PYHTON

e) COBOL e OBJECT PASCAL

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 3: Tema - 1 . PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO: MOTIVAÇÃO E PRELIMINARES

3 Objetivos

Apresentar e discutir os principais critérios de avaliação d e linguagens de programação que poderão ser usados como argumentos para escolha de uma linguagem que melhor atenda às necessidades de um problema específico

4 Tópicos

1 .3 TRADE-OFFS NO PROJETO DE LINGUAGENS

5 Procedimentos de ensino-aprendizagem

Explicar as categorias de linguagens de programação, trade-offs no projeto de linguagens, métodos de implementação. Como sugestão, segue o roteiro abaixo:

Situação-problema:
A decisão sobre a escolha da linguagem de programação a ser utilizada em um projeto leva em conta diversos aspectos, um deles diz respeito ao tempo de execução de um programa. Nesse caso, se esse requisito é importante, que tipo de linguagem de programação deveria ser escolhida levando em consideração se o programa é compilado, interpretado ou utiliza uma estratégia mista? Vamos supor que um engenheiro precisa desenvolver um software para um modem 5G o qual precisa decodificar dados a uma taxa muito alta. Nesse caso o que seria mais importante para esse projeto? o tempo de compilação desse programa ou o tempo de execução?

Metodologia:
Durante a aula o professor deverá implementar um programa em três linguagens: C/C++ (compilada), Python (interpretada) e Java (mista) de forma que o aluno verifique como o mesmo programa se comporta em cada tipo de linguagem.

Atividade verificadora de aprendizagem:
O aluno deverá implementar dois programas, um que exiba a frase "Olá Mundo" e outro que imprima os números de um 1 até 1000 e ao final deve verificar o tempo que cada um leva para ser executado nas linguagens C/C++, Python e Java. Depois deverá gerar um relatório com os resultados obtidos além disso deverá explicar a diferença entre os tempos a partir das diferenças entre as implementações

das linguagens. O relatório deve conter a descrição do hardware e do sistema operacional no qual o teste está sendo executado.

Esta atividade computará 1,0 ponto para a AV1. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 1 (Preliminares), Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2] Vídeo "Programação através de Paradigmas", disponível em https://www.youtube.com/watch?v=Pg3UeB-5FdA. É importante que o professor explique aos alunos com antecedência que eles devem. (Ative a legenda do vídeo e a traudução automática para português)

Atividade Autônoma Aura:

Questão 1: O mercado de jogos digitais movimentou mais de 1 bilhão de dólares anualmente no Brasil estando entre uns dos que mais crescem. Nesse contexto o Brasil está entre os 5 países que mais consomem conteúdo mobile no mundo. Entre as tendências do mercado mobile, estão os jogos de realidade aumentada.

Suponha que você é um programador e foi contratado por uma empresa para desenvolver um jogo que irá utilizar a realidade aumentada no qual você deve caçar e destruir o corona vírus que está espalhado pela cidade. Esse jogo utiliza um *engine* gráfica 3D para poder gerar o ambiente em realidade aumentada. Sabe-se que o processo de geração de imagens em 3D necessita de muito processamento, além disso, ele deve ser de alta velocidade para poder gerar as imagens de forma fluída. Nesse quesito, esse tipo de aplicação aproxima-se das características de uma aplicação científica. A empresa espera que você entregue o projeto do jogo em curto espaço de tempo além disso, como outros desenvolvedores poderão atuar no projeto, o código desenvolvido deve ser de entendimento simples. Sabe-se também que o programador conhece as linguagens C++ e Python, desse modo, qualquer outra linguagem utilizada no projeto implicará treinamento do mesmo. Espera-se que o jogo seja confiável e não apresente travamentos durante sua execução em um *hardware* adequado. Supondo que a empresa em questão oferece as linguagens abaixo com as características listadas,

Linguagem	Legibilidade	Compilada/Interpretada	Facilidade de Escrita
C	Baixa	Compilada	Baixa
C++	Média	Compilada	Média

Java	Média	Híbrida	Média
Python	Alta	Interpretada	Alta

A partir da análise das características das linguagens, justifique qual você escolheria para desenvolver o projeto especificado levando em consideração os requisitos pedidos.
a) C ou C++
b) C++ ou Python
c) Java ou C
d) Java ou Python
e) C++ ou Java

Questão 2: Escolher uma linguagem de programação que melhor se adapte a solução de um problema passa pela avaliação de critérios que podem ser escolhidos como premissas para esse projeto. Existem diversos critérios que podem ser levados em consideração nesse processo. Levando em consideração os critérios apresentados abaixo, faça a relação adequada entre eles e sua descrição:

1. Legibilidade

2. Facilidade de escrita

3. Custo

( ) Critério associado ao treinamento de pessoas, facilidade de escrita da linguagem, tempo de compilação de programas

( ) Uma das características ligadas a esse critério é a expressividade

( ) Critério que foi introduzido após entendimento de que as linguagens deveriam ser pensadas não do ponto de vista do computador mas sim do ponto de vista do desenvolvedor

a) 1-2-3

b) 1-3-2

c) 3-1-2

d)2-3-1

e) 3-2-1

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 4: Tema - 2 . NOMES, VINCULAÇÕES E ESCOPO

3 Objetivos

Caracterizar a natureza dos nomes e palavras especiais nas linguagens de programação, baseando-se na linguagem Python.

4 Tópicos

2 .1 VARIÁVEIS

5 Procedimentos de ensino-aprendizagem

Explicar o conceito de variável e sua importância, em seguida deve explicar suas regras de nomeação, conceito de memória e vinculação. Apresentar a linguagem de programação Python, suas características e seu uso no mercado de trabalho, além disso apresentar o ambiente de desenvolvimento, expondo os conceitos e comandos básicos a partir de apresentação de programas simples. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Computadores apenas executam tarefas que lhe são passadas por meio dos programas que são construídos a partir das linguagens de programação. Dessa forma é importante que definição de que tipo de dados estão sendo trabalhados e sua localização sejam indicados de forma correta. Suponha que um programador precisa decidir que tipo de dado uma variável que armazena valores de giroscópio, acelerômetro e barômetro que equipa um drone e sabendo que que os valores devem ser os mais precisos possíveis. Que parâmetros ele utilizará para tomar essa decisão? Qual a implicação de uma escolha equivocada?

Metodologia:
O professor deve iniciar a aula resolvendo os exercícios da aula anterior, e seguir com o debate da reportagem da BBC "As falhas numéricas que podem causar desastres" [2], depois a aula expositiva inicia com a explicação do conceito de variável e seus atributos. Para isso o professor deve apresentar como exemplo a necessidade de armazenar e manipular dados como nome de pessoas, altura, idade, data de nascimento, valores em conta bancárias, valores lógicos (verdadeiro e falso).
Nesse ponto o professor deve apresentar como declarar variáveis em C/C++ e em Python (deve apresentar que em Python o tipo de dado muda de forma dinâmica).

Atividade verificadora de aprendizagem:
O aluno deve acessar a lista de exercícios sequenciais do Site Python Brasil [3] e deve implementar os exercícios 2, 3, 7 e 9.
Esta atividade computará 1,0 ponto para a AV1. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 5 (Nomes e vinculações de escopos), páginas 198 a 215, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] "As falhas numéricas que podem causar desastres" disponível em: https://www.bbc.com/portuguese/noticias/2015/05/150513_vert_fut_bug_digital_ml#:~:text=O%20primeiro%20voo%20n%C3%A3o%20tripulado,nem%20um%20ato%20de%20sabotagem

[3] Lista de exercícios sequenciais do Site Python Brasil disponível em https://wiki.python.org.br/EstruturaSequencial e deve implementar os exercícios 2, 3, 7 e 9.

8 Aprenda +

[4] Resolver os seguintes exercícios propostos na plataforma URI ONLINE JUDGE:
- URI Online Judge | 1001 - Extremamente Básico
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1001)
- URI Online Judge | 1002 - Área do Círculo
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1002)
- URI Online Judge | 1003 - Soma Simples
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1003)

[5] Recomenda-se que os alunos acessem o site https://cognitiveclass.ai/, e realize o curso introdutório de Python for Data Science (PY0101EN), https://cognitiveclass.ai/courses/python-for-data-science/, como estudo complementar fora da sala aula

Atividade Autônoma Aura:

Questão 1: Em um sistema computacional baseado na arquitetura de Von Neumann há sempre um bloco de memória que é utilizado para armazenar valores que serão posteriormente trabalhados pela unidade de processamento ou então serão exibidos em um dispositivo de saída. Diretamente ligado a esse bloco de memória, temos o conceito de variável que pode ser entendimento como uma abstração desse bloco. Entretanto, existem alguns atributos que são utilizados para caracterizar uma variável. Qual das opções abaixo apresenta atributos relacionados com o conceito de variável?

a) nome, forma, valor, tamanho

b) tamanho, arquitetura, endereço, valor

c) arquitetura, tipo, endereço, valor

d) nome, endereço, tipo, valor

e) nome, arquitetura, tipo, valor

Questão 2: Um programador foi contratado para desenvolver um programa que utiliza um Algoritmo Genético (uma técnica de Inteligência Artificial) para encontrar a melhor rota a ser utilizada para entrega de mercadorias em cidades do Estado do Rio de Janeiro levando em consideração a menor distância percorrida (Em km entre as cidades - valores do tipo real). Esse mesmo programa também deve ser capaz de trabalhar com qualquer tipo de dado numérico, desse modo, a vinculação de tipos da linguagem escolhida para sua implementação deve ser dinâmica. Levando esse requisito em consideração, qual linguagem seria mais adequada para desenvolver esse programa?

a) Python

b) C

c) C++

d) Java

e) COBOL

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 5: Tema - 2 . NOMES, VINCULAÇÕES E ESCOPO

3 Objetivos

Entender o conceito de Escopo de uma linguagem e sua importância na manipulação de variáveis

4 Tópicos

2 .2 ESCOPO

5 Procedimentos de ensino-aprendizagem

Explicar ao aluno o conceito de escopo e sua importância, apresentar o conceito de blocos e revisitar as estruturas de decisão e estruturas de repetição utilizando a linguagem Python para exemplificar. Como sugestão, segue o roteiro abaixo:

- Situação-problema:
Muitos programadores utilizam o mesmo nome de variável em várias partes do programa cujos valores são diferentes, por exemplo, é muito comum utilizar a variável chamada i para denotar a posição de um vetor. Como é possível que elas tenham valores diferentes mesmo tendo o mesmo nome?

- Metodologia:
Aula expositiva inicia com a apresentação do conceito de escopo de uma variável, em seguida o professor deve demonstrar o conceito a partir de um programa simples em Python. Como exemplo ele pode utilizar o seguinte trecho:

```
def func():
x = 1
print(x)
x = 10
func()
print(x)
```

cujos valores impressos são 1 e 10. Nesse momento o professor deve explicar, a partir do escopo, os valores que foram impressos.
Após a apresentação do conceito de escopo, o professor deve seguir a aula apresentando as estruturas de decisão e de repetição na linguagem Python.

- Atividade verificadora de aprendizagem:

O aluno deve acessar a lista de exercícios do site Python Brasil [2] e deve implementar os exercícios 3, 5 e 7. Também deve acessar a lista de exercícios sobre estruturas de repetição [3] e implementar os exercícios 1 e 9.
Esta atividade computará 1,0 ponto para a AV1. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 5 (Nomes e vinculações de escopos), páginas 215 a 226, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Lista de exercícios de estruturas de decisão do site Python Brasil disponível em: https://wiki.python.org.br/EstruturaDeDecisao

[3] Lista de exercícios de estruturas de repetição do site Python Brasil disponível em https://wiki.python.org.br/EstruturaDeRepeticao

8 Aprenda +

[4] Resolver os seguintes exercícios propostos na plataforma URI ONLINE JUDGE:
- URI Online Judge | 1035 - Teste de Seleção 1
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1035)
- URI Online Judge | 1036 - Fórmula de Bhaskara
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1036)
- URI Online Judge | 1037 - Intervalo
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1037)
- URI Online Judge | 1038 - Lanche
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1038)
- URI Online Judge | 1041 - Coordenadas de um Ponto
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1041)
- URI Online Judge | 1043 - Triângulo
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1043)
- URI Online Judge | 1059 - Números Pares
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1059)
- URI Online Judge | 1060 - Números Positivos
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1060)
- URI Online Judge | 1064 - Positivos e Média
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1064)
- URI Online Judge | 1067 - Números Ímpares
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1067)
- URI Online Judge | 1071 - Soma de Impares Consecutivos I
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1071)

- URI Online Judge | 1072 - Intervalo 2
(https://www.urionlinejudge.com.br/judge/pt/problems/view/1072)

Atividade Autônoma Aura:

Questão 1: (QUADRIX - COBRA TECNOLOGIA, 2014)(Adaptada) Dado o programa em Python para troca de valores:

```
A = input("Digite o valor de A: ")

B = input("Digite o valor de B: ")

X = A

A = B

B = X

print("O valor de A é: ", A)

print("O valor de B é: ", B)
```

Considerando o algoritmo acima, as variáveis:

I. X, A e B foram criadas com escopo global na área de dados da memória.

II. X, A e B foram criadas com escopo locai na área de dados da memória.

III. A e B poderiam, refazendo o algoritmo, ser qualificadas com escopo local e a variável X com escopo global.

IV. A e B poderiam, refazendo o algoritmo, ser qualificadas com escopo global e a variável X com escopo local.

Está correto o que consta somente em:

a) III

b) II

c) I e III

d) I e IV

e) II e IV

Questão 2: (ENADE, 2011) Escopo dinâmico: para as linguagens com escopo dinâmico, a vinculação das variáveis ao escopo é realizada em tempo de execução. (...) Se uma variável é local ao bloco, então o uso da dada variável no bloco será sempre vinculado àquela local. Contudo, se a variável for não local, a sua vinculação depende da ordem de execução, a última vinculada na execução. A consequência disso é que, em um mesmo bloco de comandos, um identificador pode ter significados diferentes, e o programador precisa ter a ideia precisa de qual variável está sendo usada.
de MELO, A. C. V.; da SILVA, F. S. C.**Princípios de Linguagens de Programação**. São Paulo: Edgard Blücher, 2003. p.65.

Suponha que uma linguagem de programação tenha sido projetada com vinculação e verificação estáticas para tipos de variáveis, além de passagem de parâmetros por valor. Também é exigido pela especificação da linguagem que programas sejam compilados integralmente e que não é permitido compilar bibliotecas separadamente. Durante uma revisão da especificação da linguagem, alguém

propôs que seja adicionado um mecanismo para suporte a variáveis com escopo dinâmico.

A respeito da proposta de modificação da linguagem, analise as seguintes afirmações.

I. As variáveis com escopo dinâmico podem ser tratadas como se fossem parâmetros para os subprogramas que as utilizam, sem que o programador tenha que especificá-las ou declarar seu tipo (o compilador fará isso). Assim, eliminase a necessidade de polimorfismo e é possível verificar tipos em tempo de compilação.

II. Como diferentes subprogramas podem declarar variáveis com o mesmo nome mas com tipos diferentes, se as variáveis com escopo dinâmico não forem declaradas no escopo onde são referenciadas, será necessário que a linguagem suporte polimorfismo de tipos.

III. Se as variáveis dinâmicas forem declaradas tanto nos escopos onde são criadas como nos subprogramas em que são referenciadas, marcadas como tendo escopo dinâmico, será possível identificar todos os erros de tipo em tempo de compilação.

É correto apenas o que se afirma em:

a) I.

b) II.

c) I e III.

d) II e III.

e) I, II e III.

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 6: Tema - 3. TIPOS DE DADOS

3 Objetivos

Especificar variáveis, empregando tipos de dados, de forma a contextualizar ao compilador/interpretador como o programador pretende utilizar os dados.

4 Tópicos

3.1 PRIMITIVOS

5 Procedimentos de ensino-aprendizagem

Apresentar os tipos primitivos de dados e cadeias de caracteres, e apresentar enumerados e matrizes sempre apontando situações nas quais são aplicados e apresentando exemplos na linguagem Python. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Computadores precisam armazenar dados e os tipos de dados armazenados devem refletir, muitas vezes, algo do mundo real. As linguagens de programação devem manipular esses dados e elas muitas vezes trabalham com os dados chamados de primitivos como números inteiros, reais, caracteres e lógicos. Suponha que um programador necessite armazenar dados do mesmo tipo relativos a temperatura de uma cidade a cada hora do dia conforme tabela abaixo:
Dia/hora 08:00 12:00 16:00 20:00
01/01/2020 25.0 32.5 30.5 28.0
02/01/2020 26.0 31.6 32.5 29.5
Pergunta: A utilização de dados do tipo primitivo seria suficiente?

Metodologia:
Aula expositiva inicia com a apresentação dos tipos primitivos de dados seguidos da apresentação de vetores, matrizes, cadeias de caracteres e formas de manipulação dos mesmos. Exemplos práticos devem ser apresentados utilizando a linguagem de programação Python.

Atividade verificadora de aprendizagem:
O aluno deve acessar a lista de exercícios sequenciais do Site Python Brasil [2] e implementar os exercícios 1, 3 e 5, também deve acessar a lista de exercício sobre listas [3] e implementar os exercícios 1 e 6.

Os exercícios devem ser resolvidos pelo professor durante o tempo de aula e não havendo tempo, devem ser resolvidos na aula seguinte.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 6 (Tipos de dados), páginas 234 a 262, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Lista de exercícios com Strings do Site Python Brasil disponível em https://wiki.python.org.br/ExerciciosComStrings

[3] Lista de exercícios com Listas do Site Python Brasil disponível em https://wiki.python.org.br/ExerciciosListas

8 Aprenda +

[4] Vídeo "Matrizes". Disponível em: https://www.youtube.com/watch?v=c9yjwWNiNQw&list=PLcoJJSvnDgcKpOi_UeneTNTIVOigRQwcn&index=32&t=0s

-

[5] Vídeo "Strings". Disponível em: https://www.youtube.com/watch?v=DdhNltkI_hE&list=PLcoJJSvnDgcKpOi_UeneTNTIVOigRQwcn&index=33

Atividade Autônoma Aura:

Questão 1:(FGV-TJ_BA, 2015)(Adaptada) Analise o trecho de programa, escrito em Pyhton na versão 2.7, mostrado a seguir.

```
X=[ ]

for i in range(10,1,-1):

X.append(i)

print (X[3:])
```

O trecho exibe:

a) [7, 6 ,5]

b) [7]

c) []

d) [7,6,5,4,3,2]

e) [7,6,5,4,3,2,1]

Questão 2: Considere o seguinte trecho de programa em Python

```
Nota1 = input("Digite a primeira nota: ")
Nota2 = input("Digite a segunda nota: ")
media = (Nota1+Nota2)/2
print("A média é: ",media)
```

Qual a saída desse programa supondo que o usuário digitou 10 e 7?

a) 8.5

b) 8

c) "107"

d) "8.5"

e) É apresentada uma mensagem de erro de tipo

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 7: Tema - 3 . TIPOS DE DADOS

3 Objetivos

Apresentar estruturas que podem comportar dados de diversos tipos e operações sobre essas estruturas.

4 Tópicos

3.2 AGLOMERADOS

5 Procedimentos de ensino-aprendizagem

Apresentar o conceito de Registros, demonstrar como é implementado em C/C++ Struct, Tuplas, Listas e Uniões. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Em muitas situações é necessário que dados sejam organizado em coleções que NÃO são do mesmo tipo ou tamanho, como por exemplo, informações sobre um estudante universitário que incluem seu nome, seu número de matrícula e média. Não podemos utilizar uma matriz para realizar esse armazenamento pois os dados nesse caso são de tipos diferentes. Que tipo de estrutura você utilizaria?

Metodologia:
Aula expositiva inicia com a apresentação de Registros que podem ser implementados em C/C++ a partir de struct. Nesse caso o professor deve apresentar um exemplo prático de implementação, após o professor deve apresentar o conceito de Tuplas e Listas aos alunos utilizando a linguagem Python.

Atividade verificadora de aprendizagem:
Utilizando listas faça o aluno deve implementar um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:
a. "Telefonou para a vítima?"
b. "Esteve no local do crime?"
c. "Mora perto da vítima?"
d. "Devia para a vítima?"
e. "Já trabalhou com a vítima?" O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele

será classificado como "Inocente".

O exercício deve ser resolvido pelo professor durante o tempo de aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 6 (Tipos de dados), páginas 263 a 287, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2] Resolver o exercício "URI Online Judge | 1088 - Bolhas e Baldes", disponível em https://www.urionlinejudge.com.br/judge/pt/problems/view/1088

Atividade Autônoma Aura:

Questão 1: Dado seguinte trecho de código em Python:

```
programadores = ['Victor', 'Juliana', 'Samuel', 'Caio', 'Luana']
print(type(programadores))
print(len(programadores))
print(programadores[4])
```

Identifique o tipo de estrutura que está sendo utilizada:

a) Lista
b) Tupla
c) Struct
d) Matriz
e) Hash

Questão 2: Dado o seguinte trecho de código em Python:

```
programadores = ['Victor', 'Juliana', 'Samuel', 'Caio', 'Luana']
programadores.insert('Rafael')
print(programadores)
```

Análise as afirmativas a seguir:

I - Ao utilizar o comando insert, o elemento especificado será inserido no final da estrutura.
II - O comando insert utiliza dois parâmetros: a posição a ser inserida seguida do elemento.

III - Se insert for substituído por append o elemento será inserido na estrutura.

Qual(s) afirmativa(s) é(são) correta(s)?

a) I e II
b) I
c) II e III
d) I e III
e) III

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 8: Tema - 3 . TIPOS DE DADOS

3 Objetivos

Empregar ponteiros para construir estruturas que realizem a alocação de memória de forma dinâmica.

4 Tópicos

3.3 PONTEIROS E REFERÊNCIAS
3.4 VERIFICAÇÃO DE TIPOS
3.5 TEORIA E TIPOS DE DADOS
3.6 DADOS ABSTRATOS E ENCAPSULAMENTO

5 Procedimentos de ensino-aprendizagem

A aula trata dos conceitos de ponteiros e referências e operações, aplicando o conceito de alocação dinâmica de memória na implementação de estruturas como listas encadeadas, pilhas e filas utilizando para isso a linguagem C/C++. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Suponha que um programador é contratado para desenvolver um programa que precisa armazenar o nome e a média de todos os alunos de uma universidade mas não sabe quantos alunos ela possui, o que impede que ele declare previamente a quantidade de memória necessária para armazenar esses dados. Nesse caso ele necessita realizar uma alocação de memória de forma dinâmica. Como ele faria isso?

Metodologia:
Aula expositiva inicia com a apresentação do conceito de ponteiros e sua utilização na linguagem de programação C/C++. O professor deve apresentar a sua utilização quando há a necessidade de acessar uma variável em diversos pontos de um programa, quando há a necessidade de realizar alocação dinâmica de memória fazendo referência a lista e pilhas. Nesse ponto, o professor deve implementar uma lista encadeada de forma que os alunos apliquem o conceito de ponteiros em uma estrutura da de dados.

Atividade verificadora de aprendizagem:
A partir do exemplo da lista encadeada demostrada em aula pelo professor. Os alunos devem ser divididos em duplas e devem implementar os métodos de inserção e remoção de um elemento da lista. De forma a tornar a atividade mais simples, a lista deve conter apenas elementos do tipo inteiro.

O exercício deve ser resolvido pelo professor durante o tempo de aula e não havendo tempo, deve ser resolvido na aula seguinte.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 6 (Tipos de dados), páginas 263 a 287, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2] Vídeo "Listas encadeadas". Disponível em https://www.youtube.com/watch?v=njTh_OwMljA (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (COTEC - Prefeitura de Japonvar MG, 2013) (Adaptada) A linguagem C permite alocar (reservar) dinamicamente (em tempo de execução) blocos de memórias utilizando ponteiros. A esse processo dá-se o nome de alocação dinâmica, que faz uso das funções malloc, calloc, realloc e free, disponíveis na biblioteca stdlib.h. Para liberar um bloco de memória previamente alocado, por meio de um único parâmetro de entrada, faz-se uso de qual função?

a) malloc

b) calloc

c) realloc

d) free

e) alloc

Questão 2: (FCC - TRT - AM, 2010) Na linguagem C++, considere:

I. O endereço armazenado em um ponteiro deve ser do mesmo tipo que o ponteiro (ex. um ponteiro para um int não pode armazenar o endereço de um float).

II. Exceção à regra apontada em (I) é o ponteiro void.

III. Não é possível chamar uma função segundo seu endereço, ainda que por meio de um ponteiro que armazena o endereço de início dessa função.

Está correto o que se afirma em

a) I, apenas.

b) II, apenas.

c) I e II, apenas.

d) II e III, apenas.

e) I, II e III.

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 9: Tema - 4 . EXPRESSÕES E SENTENÇAS DE ATRIBUIÇÃO

3 Objetivos

Empregar formas fundamentais de instruções, baseando-se na sintaxe e semântica de expressões aritméticas, relacionais e booleanas e atribuições, para escrever instruções matemática e lógicas compreensíveis, corretas e executáveis por computadores.

4 Tópicos

4 .1 INTRODUÇÃO À EXPRESSÕES E SENTENÇAS DE ATRIBUIÇÃO
4 .2 SENTENÇAS DE ATRIBUIÇÃO

5 Procedimentos de ensino-aprendizagem

O professor deve iniciar a aula apresentando situações nas quais seja possível estabelecer relação entre o tópico desta aula e o objetivo. Como sugestão, segue o roteiro abaixo:

Situação-problema:
A precedência de operadores é um conceito matemático que determina a prioridade de certos operadores sobre outros no momento da realização. Atualmente circulam na internet diversos desafios para que os internautas forneçam o resultado de operações matemáticas como por exemplo
8 / 2(2+2) = ?
Nesse caso a podemos ter duas soluções distintas? Qual o motivo?

Metodologia:
O professor deve iniciar a aula apresentando a importância da utilização de expressões aritméticas, relacionais e booleanas no contexto de problemas reais. Deve apresentar as possibilidades de conversão de tipos. Além disso, o aluno deve compreender a operação de atribuição e suas diversas possibilidades como a atribuição unária, atribuição de listas, atribuição como uma expressão e atribuição de modo misto. Ao final de cada tópico, o professor deve resolver exercícios.

Atividade verificadora de aprendizagem:
Analise o programa abaixo e, para cada uma das saídas (comandos print), detalhe passo a passo como o Python (segundo suas prioridades) resolveria as equações e o resultado final obtido.
x = 2
y = 3

```
z = 0.5
print(x + x * x ** (y * x) / z)
print(not x + z < y or x + x * z >= y and True)
```

Faça um programa Python que calcule a área do cubo.
Escreva um programa que recebe três inteiros como entrada do teclado e escreva na tela a média, a soma, o produto, o menor valor e o maior valor, usando uma linha para cada resultado.
Os exercícios devem ser resolvidos pelo professor durante o tempo de aula e não havendo tempo, devem ser resolvidos na aula seguinte.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 6 (Tipos de dados), páginas 301 a 322, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2]Documentação "Operadores aritméticos e booleanos". Disponível em https://docs.python.org/pt-br/3/library/stdtypes.html

Atividade Autônoma Aura:

Questão 1: Analise o seguinte trecho de código:

```
n1 = int(input('Primeiro numero: '))

n2 = int(input('Segundo numero : '))

n3 = int(input('Terceiro numero: '))

print(n1,'-',n2,'-',n3)

if(n3 > n2):

aux = n3

n3 = n2

n2 = aux

if(n2 > n1):

aux = n2

n2 = n1
```

```
n1 = aux

if(n3 > n2):

aux = n3

n3 = n2

n2 = aux

print(n1,'-',n2,'-',n2)
```

a) O programa irá ordenar os números em ordem crescente

b) O programa irá ordenar os números em ordem decrescente

c) O programa apresentará um erro no comando input

d) O programa apresentará um erro pois está utilizando a variável aux de forma global

e) O programa apresentará erro caso o usuário entre com números iguais

Questão 2: (NC-UFPR ? ITAIPU BINACIONAL, 2017) Sejam os seguintes comandos python executados na sequência apresentada:

```
x = range(10)
def somar(x,y):
        return x+y
x = [i**2 for i in x if i%2 == 0]
reduce(somar,x)
```

Qual é o resultado da execução?

a) 285

b) 120

c) 90

d) 45

e) 20

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 10: Tema - 5 . SUBPROGRAMAS

3 Objetivos

Empregar modularização de programas para decompor problemas complexos em partes menores favorecendo a solução completa.

4 Tópicos

5 .1 FUNDAMENTOS DOS SUBPROGRAMAS

5 Procedimentos de ensino-aprendizagem

O professor deve iniciar a aula apresentando situações nas quais seja possível estabelecer relação entre o tópico desta aula e o objetivo. Como sugestão, segue o roteiro abaixo:

Situação-problema:
Suponha que um programador é contrato para desenvolver um sistema ERP (Enterprise Resource Planning) que é um software integrado para gestão empresarial formado por diversos módulos como: Contabilidade, Recursos Humanos, Suprimentos, Patrimônio, Vendas entre outros. Desenvolver um sistema desse porte em um único arquivo que contém todo o programa seria extremamente complexo tanto para sua manutenção como para o próprio entendimento haja vista a grande quantidade de código sequencial que o mesmo teria. Como poderíamos resolver esse problema dando mais organização para o esse programa?

Metodologia:
O professor deve iniciar a aula apresentando ao aluno os fundamentos de subprogramas, demonstrando a importância da segmentação de problemas grandes em pequenos problemas logicamente encadeados. O professor deve apresentar a sintaxe da construção de funções na linguagem de programação Python além de apresentar os métodos de passagem de parâmetros. Deve-se apresentar um exemplo de programa recursivo, como o cálculo de fatorial. O professor também deverá implementar exemplos de subprogramas utilizando a linguagem de programação Python.

Atividade verificadora de aprendizagem:
Faça um programa com uma função chamada somaImposto. A função possui dois parâmetros formais: taxaImposto, que é a quantia de imposto sobre vendas expressa em porcentagem e custo, que é o custo

de um item antes do imposto. A função "altera" o valor de custo para incluir o imposto sobre vendas.

Faça um programa que converta da notação de 24 horas para a notação de 12 horas. Por exemplo, o programa deve converter 14:25 em 2:25 P.M. A entrada é dada em dois inteiros. Deve haver pelo menos duas funções: uma para fazer a conversão e uma para a saída. Registre a informação A.M./P.M. como um valor "A" para A.M. e "P" para P.M. Assim, a função para efetuar as conversões terá um parâmetro formal para registrar se é A.M. ou P.M. Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar.

Esta atividade computará 1,0 ponto para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 9 (Subprogramas), páginas 364 a 393, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Documentação do Python "Principios da Programação Funcional em Python". Disponível em https://wiki.python.org.br/PrincipiosFuncionais

[3] Listas de exercícios do Python Brasil sobre Funções, disponível em: https://wiki.python.org.br/ExerciciosFuncoes

8 Aprenda +

[4] Vídeo "Recursão". Disponível em https://www.youtube.com/watch?v=KEEKn7Me-ms (Ative a legenda e a tradução automática para o português)

-

[5] Vídeo "Como usar funções no Python (Tutorial do Python # 3)". Disponível em https://www.youtube.com/watch?v=NSbOtYzIQI0 (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (IF Sul Riograndense, 2019) (adaptada)Observe a função print do código escrito em Python a seguir:

```
def foo(n):
if n> 1:
return n * foo(n-1)
```

print(foo(4))

Qual o resultado impresso por essa função?

a) 4
b) 16
c) 24
d) 20
e) 48

Questão 2: (AOCP - MJSP, 2020) Assinale a alternativa que apresenta o código Python que implementa corretamente uma função para o cálculo da área de um retângulo, bem como o código que imprime o seu resultado.

a) def area_retangulo(comp, larg){
return comp * larg}
print(' A área é ' +
area_retangulo (2,4))

b) func area_retangulo(comp, larg):return comp * largprint(' A área é ' ||
area_retangulo (2,4))

c) def area_retangulo(comp, larg)(
return comp * larg);
print(' A área é {}' .format (area_retangulo (2,4)))

d) func area_retangulo(comp, larg):return comp * larg;println(' A área é ' +
area_retangulo (2,4))

e) def area_retangulo(comp, larg):
return comp * larg
print(' A área é {}' .format
(area_retangulo (2,4)))

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 11: Tema - 5 . SUBPROGRAMAS

3 Objetivos

Utilizar bibliotecas na linguagem Python potencializando as possibilidades de uso da linguagem.

4 Tópicos

5 .2 QUESTÕES DE PROJETO PARA SUBPROGRAMAS

5 Procedimentos de ensino-aprendizagem

A aula tratará do conceito de biblioteca e como utilizá-la na linguagem Python. Como sugestão, segue o roteiro:

Situação-problema:
Imagina que um programador precisa desenvolver um software que exiba o gráfico da evolução do número de infectados por uma doença e seus parâmetros estatísticos como média móvel, variância e desvio padrão. Esse programador teria que se preocupar em desenvolver muitos recursos, desde a estrutura de armazenamento dos dados passando pelo módulo estatístico, que utiliza muitas funções matemáticas, até o módulo que irá desenhar o gráfico da função. Será que essas funcionalidades já não foram desenvolvidas por outros programadores? Como podemos utilizá-las em nossos próprios projetos?

Metodologia:
Aula expositiva inicia com o conceito de biblioteca e sua utilidade, com base na documentação "Bibliotecas Padrão do Python" [2]. Após esse momento, o professor deve apresentar como utilizar bibliotecas na linguagem de programação Pyhton e posteriormente deve apresentar e implementar exemplos com as principais bibliotecas disponíveis para a linguagem de programação Python.

Atividade verificadora de aprendizagem:
Suponha que o programador necessite desenvolver uma calculadora científica que deve possuir, além das operações aritméticas básicas (adição, subtração, divisão e multiplicação), a possibilidade de calcular raiz quadrada, potenciação de números, funções trigonométricas (seno, cosseno, tangente). Qual biblioteca você utilizaria e quais funções você deveria utilizar para a implementação da calculadora?

Esta atividade computará 1,0 ponto para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 9 (Subprogramas), páginas 364 a 393, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Documentação Python "Bibliotecas Padrão do Python". Disponível em https://docs.python.org/pt-br/3/library/.

8 Aprenda +

[3] Documentação do Python "Princípios da Programação Funcional em Python". Disponível em https://wiki.python.org.br/PrincipiosFuncionais

Atividade Autônoma Aura:

Questão 1: (FCC - SANASA Campinas, 2019) Uma Analista de TI está desenvolvendo um *scanner* de rede em Python e, para importar um recurso referente para manipulação de pacotes de rede, utilizou no programa a linha

a) import pyTTS *

b) from Tkinter import *

c) import aiml *

d) from scapy.all import *

e) from numpy import *

Questão 2: (FCC-TRE SP,2017) Considere o programa Python, abaixo.

```
import _________ as b
import matplotlib.pyplot as a
x = b.linspace(0, 3, 20)
y = b.linspace(0, 9, 20)
a.plot(x, y)
a.plot(x, y, 'o')
```

a.show()

A lacuna _____ deve ser preenchida corretamente com
a) numpy
b) matrix
c) mathlab
d) numberplot
e) array

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 12: Tema - 5 . SUBPROGRAMAS

3 Objetivos

Construir programas complexos a partir de subprogramas e sua interações

4 Tópicos

5.2 QUESTÕES DE PROJETO PARA SUBPROGRAMAS

5 Procedimentos de ensino-aprendizagem

O professor deve iniciar a aula apresentando situações nas quais seja possível estabelecer relação entre o tópico desta aula e o objetivo. Como sugestão, segue o roteiro abaixo:

Situação-problema:
O desenvolvimento de softwares complexos e grandes demanda que eles sejam divididos em partes menores. Uma das formas de implementar essas pequenas partes é por meio de subprogramas. Você enxerga outra possibilidade?

Metodologia:
Nessa aula o professor deve apresentar, por meio de exemplos, a semântica geral de chamadas e retornos de funções aliando sempre essa tarefa a utilização de bibliotecas sendo que os alunos devem implementar os exemplos junto com o professor.

Atividade verificadora de aprendizagem:
O professor deve iniciar a aula apresentando ao aluno a semântica geral de chamadas e retorno e após deve apresentar exemplos de subprogramas simples, conforme documentação do Python "Interfaces Gráficas de Usuário com Tkinter" [2]. Utilizando a biblioteca gráfica Tkinter, o aluno deve construir uma interface que simule uma tela de login de usuário contendo o campo de login, senha e um botão de autenticação

Esta atividade computará 1,0 ponto para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

Além disso o professor deve solicitar aos alunos que estudem o código de uma calculadora

implementada em Python [3]. Como projeto opcional, os alunos devem incluir funções trigonométricas no projeto da calculadora.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 9 (Subprogramas), páginas 364 a 393, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Documentação do Python "Interfaces Gráficas de Usuário com Tkinter", disponível em https://docs.python.org/pt-br/3/library/tk.html

[3] Calculadora implementada no site Python Brasil, disponível em: https://wiki.python.org.br/CalculadoraSimplesTk

8 Aprenda +

[4] Documentação do Python "Principios da Programação Funcional em Python". Disponível em https://wiki.python.org.br/PrincipiosFuncionais

Atividade Autônoma Aura:

Questão 1 (CESGRANRIO - UNIRIO, 2019) O código a seguir exibe parte de um programa Python que tem por objetivo retirar um elemento de uma pilha (variável stack) de strings e exibir no console o valor do elemento retirado.

```
def empty(s):
if len(s)==0:
return True
else:
return False
def pop(s):
#falta a implementação
#desta função
stack = ["calça","meias","paletó","gravata","camisa"]
if not empty(stack):
print(pop(stack))
else:
print("Pilha vazia")
```

A pilha foi concebida de modo que o seu topo é o primeiro elemento de uma lista (variável stack). Qual versão da função pop(s) fará com que o programa acima alcance o seu objetivo?

a) def pop(s): return s.pop()
b) def pop(s): temp=s[0] s.remove(s[0]) return temp
c) def pop(s): return s.pop(-1)
d) def pop(s): temp=s[0] s=s[1:] return temp

e) def pop(s): temp=s[-1] s=s[-1:1:-1] return temp

Questão 2: (CESGRANRIO -FINEP, 2014) Uma linguagem de programação permite que os parâmetros de uma função sejam passados por valor ou por referência. Suponha que nessa linguagem seja definida uma função F(A,B) onde A e B são os parâmetros formais, sendo que A é passado por valor, e B é passado por referência. Durante a execução de F, somamos 2 ao valor de A e subtraímos 2 do valor de B.
Caso F(X,Y) seja uma chamada da função, ao longo do programa, onde os parâmetros reais X e Y são variáveis cujos valores antes da chamada são, respectivamente, 10 e 20, esperamos que, ao terminar a função, os novos valores de X e Y sejam, respectivamente,
a) 10 e 18
b) 10 e 20
c) 10 e 22
d) 12 e 18
e) 12 e 20

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 13: Tema - 6. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO (ATIVIDADE PRÁTICA SUPERVISIONADA)

3 Objetivos

Apresentar o paradigma orientado a objetos e sua importância para a modelagem de sistemas

4 Tópicos

6.1 PARADIGMA ORIENTADO A OBJETO

5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual. Durante a aula, este conteúdo será discutido em sala em atividade mediada pelo professor, detalhada abaixo.

Situação-problema:
Os paradigmas de programação são um modelo, uma metodologia de programação. Você já conhece o paradigma conhecido como programação estruturada que utiliza subprogramas e três estruturas básicas: sequência, seleção e repetição. Seus códigos ficam em um mesmo bloco, sendo mais difícil e demorado realizar alterações pois é necessário verificar se outro trecho de código depende dele. Vamos supor que um determinado software possui uma função de conexão com um banco de dados, a qual é utilizada em diversos pontos dele e que é utilizada de forma corriqueira em outros softwares. Como você poderia reutilizar esse código? De que forma poderia ser disponibilizado?

Metodologia:
Aula inicia com um rápido debate sobre o entendimento do paradigma orientado a objetos em comparação ao paradigma estruturado. Em seguida o professor deve propor uma atividade prática de implementação de uma classe utilizando a linguagem de programação Python.
Classe Pessoa: Crie uma classe que modele uma pessoa:
a. Atributos: nome, idade, peso e altura
b. Métodos: Envelhecer, engordar, emagrecer, crescer.
Obs: Por padrão, a cada ano que nossa pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 0,5 cm.
Após a implementação da classe o professor deve indagar aos alunos como eles iriam implementar em um paradigma estruturado essa mesma classe. Nesse ponto o professor deve fazer um paralelo entre

funções e métodos, variáveis e objetos, chamadas a funções e mensagens.

Atividade verificadora de aprendizagem:
Classe Bichinho Virtual: Crie uma classe que modele um Tamagoshi (Bichinho Eletrônico): (Exercício 7 da lista de exercícios sobre Classes do site Python Brasil [3]).
a. Atributos: Nome, Fome, Saúde e Idade
b. Métodos: Alterar Nome, Fome, Saúde e Idade; Retornar Nome, Fome, Saúde e Idade e imprimir (deve apresentar os valores de todos os atributos)
Após a criação da classe, instancie dois Tamagoshis e altere seus atributos e depois os imprima.
Como atividades opcionais o aluno pode implementar os demais exercícios disponíveis no site Python Brasil [3]

Esta atividade computará 2,0 pontos para a AV2. O docente deve acompanhar o desenvolvimento dos alunos, dirimindo dúvidas, e zelando pelo curretude das soluções. Recomenda-se o uso do SAVA ou Teams para a realização das entregas.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 12 (Suporte para programação orientada a objetos), páginas 488 a 512, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Documentação do Python "Python e Programação Orientada a Objeto". Disponível em https://wiki.python.org.br/ProgramacaoOrientadaObjetoPython

[3] Exercício 7 da lista de exercícios sobre Classes do site Python Brasil, disponível em https://wiki.python.org.br/ExerciciosClasses

8 Aprenda +

[4] Vídeo "Programação orientada a objetos". Disponível em: https://www.youtube.com/watch?v=lbXsrHGhBAU (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (FUNCERN - IF-RN, 2017)(Adaptada)
Analise o seguinte código escrito em Python, que define a estrutura da classe ContaBancaria:

```
class ContaBancaria:

num_contas = 0

def __init__(self, agencia, numero, saldo):

self.agencia=agencia
```

```
self.numero=numero

self.saldo=saldo

ContaBancaria.num_contas +=1

def __del__(self):

ContaBancaria.num_contas -=1

def depositar (self,valor):

self.saldo = self.saldo + valor

def sacar (self,valor):

self.saldo = self.saldo ? valor

def consultarSaldo(self):

return self.saldo
```

Sobre a classe acima e as regras de programação orientada a objetos em Python,

a) criação de objetos chama primeiro o método __init__() e, em seguida, o __new__().

b) palavra self deve ser fornecida como argumento em todos os métodos de instâncias.

c) variável num_contas é encapsulada e individual para cada instância da classe.

d) palavra @static escrita antes da definição do método sacar() torna o método estático.

e) estamos diante de situação onde há herança

Questão 2: (SUGEP-UFRPE, 2016) Considere as afirmações abaixo, sobre os paradigmas de linguagens de programação.
1) As linguagens de programação Python, Ruby, C#, Cython e Lua são multiparadigmáticas e podem ser classificadas, pelo menos, nos paradigmas Orientado a Objetos, Funcional e Imperativo.
2) As linguagens de programação Object-Pascal (Delphi), Python, C++ e Java, embora deem suporte à Orientação a Objetos (OO), não são completamente orientadas a objetos.
3) As linguagens de programação Smalltalk e Ruby são completamente orientadas a objetos, uma vez que todo valor de dados é um objeto e todas as operações são vias chamadas de métodos.
Está(ão)correta(s):

a) 1, 2 e 3.

b) 2 e 3, apenas.

c) 1 e 2, apenas.

d) 1 e 3, apenas.

e) 3, apenas.

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 14: Tema - 6. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO (ATIVIDADE PRÁTICA SUPERVISIONADA)

3 Objetivos

Aplicar e apresentar os conceitos de polimorfismo, herança, encapsulamento na orientação a objeto

4 Tópicos

6.1 PARADIGMA ORIENTADO A OBJETO

5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual. Durante a aula, este conteúdo será discutido em sala em atividade mediada pelo professor, detalhada abaixo.

Situação-problema:
Os softwares muitas vezes possuem partes que são repetidos em várias situações. Vamos supor que você criou um módulo de impressão que é utilizado em diversos softwares. Como você poderia reutilizar esse módulo sem ter que copiar o código dele para programas diversos?

Metodologia:
Aula com um rápido debate sobre herança e polimorfismo discutindo as vantagens de utilização desses dois princípios da orientação a objetos, conectando com a situação problema apresentada. Em seguida o professor deve implementar a seguinte atividade junto com alunos para demonstrar o conceito de herança.
Criar a classe Pessoa
a. Atributos: Nome, Endereço
b. Métodos: Alterar Nome, Endereço; Retornar Nome, Endereço.
Em seguida deve criar a classe Aluno que irá herdar os atributos e métodos da classe Pessoa e a irá acrescentar
a. Atributo: Nota
b. Métodos: Alterar e Retornar Nota
Devem ser implementados na linguagem Python e na Linguagem Java.

Atividade verificadora de aprendizagem:

Considere, como subclasse da classe Pessoa a classe Fornecedor. Considere que cada instância da classe Fornecedor tem, para além dos atributos que caracterizam a classe Pessoa, os atributos valorCredito (correspondente ao crédito máximo atribuído ao fornecedor) e valorDivida (montante da dívida para com o fornecedor). Implemente na classe Fornecedor, para além dos usuais métodos seletores e modificadores, um método obterSaldo() que devolve a diferença entre os valores dos atributos valorCredito e valorDivida. Depois de implementada a classe Fornecedor, crie um programa de teste adequado que lhe permita verificar o funcionamento dos métodos implementados na classe Fornecedor e os herdados da classe Pessoa.
O professor deve apresentar a solução da atividade durante a aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 12 (Suporte para programação orientada a objetos), páginas 536 a 540, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

[2] Documentação do Python "Python e Programação Orientada a Objeto". Disponível em: https://wiki.python.org.br/ProgramacaoOrientadaObjetoPython

8 Aprenda +

[3] Vídeo "Programação orientada a objetos". Disponível em: https://www.youtube.com/watch?v=lbXsrHGhBAU (Ative a legenda e a tradução automática para o português)

Atividade Autônoma Aura:

Questão 1: (FGV - DPE-RJ, 2014) Considere o seguinte trecho de um programa escrito na linguagem Python.

```
class Carro(object):
def FaleComigo(self):
print "Sou um carro"

class Fusca (Carro):
def FaleComUmFusca(self):
print "Sou um Fusca"

x = Carro( )
y = Fusca( )

x.FaleComigo( )
y.FaleComigo( )
```

No primeiro bloco, o método*FaleComigo*é definido para a classe*Carro*, que simplesmente produz a mensagem "Sou um carro" ao ser invocado. Para a classe*Fusca*, definida no segundo bloco, foi

definido o método*FaleComUmFusca*, que apenas produz a mensagem "Sou um Fusca". No terceiro bloco, os objetos x e y tornam-se instâncias das classes*Carro*e*Fusca*, respectivamente. No quarto bloco, o método*FaleComigo*é invocado para cada um dos dois objetos, x e y. Ao ser executado, esse programa produz duas linhas na sua tela de saída:

Sou um carro
Sou um carro

A mensagem produzida no comando y.*FaleComigo*deve-se ao mecanismo de

a) abstração

b) associação

c) interface

d) herança

e) polimorfismo

Questão 2: (COMPERVE - UFRN, 2016) (Adaptada) Sobre a implementação de conceitos de programação orientada a objetos na linguagem Python, é correto afirmar:

a) para construir uma hierarquia de classes, deve-se passar como parâmetro na definição da classe o construtor*self*.

b) ao aplicar o conceito de encapsulamento, não se deve declarar os métodos*get/set*como única estratégia de exposição dos atributos.

c) uma classe que implementa uma interface não é obrigada a implementar todos os métodos definidos na interface.

d) para definir um construtor em uma classe em Python, utiliza-se o método especial _init_, passando como parâmetro a referência da classe.

e) Pyhton não suporta herança múltipla

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 15: Tema - 6. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO (ATIVIDADE PRÁTICA SUPERVISIONADA)

3 Objetivos

Apresentar o paradigma de programação funcional e as principais linguagens de programação que o utilizam.

4 Tópicos

6.2 PARADIGMA FUNCIONAL

5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual. Durante a aula, este conteúdo será discutido em sala em atividade mediada pelo professor, detalhada abaixo.

Situação-problema:
Imagine que você utiliza uma linguagem de programação com suporte aos principais paradigmas de programação e precisa resolver o seguinte problema:
Desenvolver um software que receba uma lista de números inteiros e retorne uma lista contendo somente os valores pares.
Você pode utilizar a sequência de passos abaixo:
1. Descobrir o tamanho da lista de números recebida
2. Inicializar uma lista vazia
3. Percorrer todos os elementos da lista
4. Para cada elemento, verificar se o valor é par
5. Caso seja par, adicionar o elemento a uma nova lista
6. Retornar a lista com os valores pares

Agora imagine uma solução onde apenas o seu objetivo é declarado

return lista.filtro(par)

Ambas as soluções possuem o mesmo resultado. Qual a proposta parece ser mais expressiva e reutilizável?

Metodologia:
Aula inicia com um rápido debate sobre as principais linguagens de programação funcional e sua comparação com o paradigma estruturado trabalhando as vantagens e desvantagens de cada paradigma. Em seguida o professor deve apresentar o conceito de recursão utilizando o exemplo de cálculo do fatorial de uma função no qual ele implementa em Python o algoritmo de forma imperativa e de forma recursiva.

Atividade verificadora de aprendizagem:
A série de Fibonacci é uma sequência de números, cujos dois primeiros são 0 e 1. O termo seguinte da sequência é obtido somando os dois anteriores. Faça uma script em Python que solicite um inteiro positivo ao usuário, n. Então uma função exibe todos os termos da sequência até o n-ésimo termo. Use recursividade.
O professor deve apresentar a solução da atividade durante a aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 12 (Suporte para programação orientada a objetos), páginas 680 a 721, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2] Artigo "Recursão". Disponível em https://panda.ime.usp.br/pensepy/static/pensepy/12-Recursao/recursionsimple-ptbr.html

Atividade Autônoma Aura:
Questão 1: (FCC - SEFAZ-SP, 2013) Para a programação dosoftware, a equipe de TI contratada pelo Sr. Hiroshito pretende adotar um paradigma de programação e uma linguagem que suporte tal paradigma. Para isso, conduziu uma pesquisa sobre os principais paradigmas e linguagens de programação. A pesquisa revelou diversos paradigmas, mas foram selecionados apenas o estruturado, o funcional e o orientado a objetos. Selecionou-se, então, uma ou mais linguagens que suportam cada paradigma.

Estão relacionados corretamente uma ou mais linguagens de programação ao respectivo paradigma de programação suportado em:

a) **Estruturado -**Cobol e Lua
Funcional -C#
Orientado a objetos -Ruby

b) **Estruturado -**Pascal e Cobol
Funcional -LISP
Orientado a objetos -Ruby e C#

c) **Estruturado -**C++

Funcional -LISP e Cobol
Orientado a objetos -Ruby e C

d) **Estruturado -**Ruby e C++
Funcional -Lua e Prolog
Orientado a objetos -Pascal e Java

e) **Estruturado -**Pascal e Delphi
Funcional -Lua
Orientado a objetos -Ruby e C#

Questão 2: (CETAP - MPC-PA, 2015) Qual das definições a seguir pertence ao paradigma funcional de linguagens programação?

a) Operandos em expressões são enviados da memória para a CPU, e o resultado da avaliação dessas expressões é transferido da CPU para a memória, representada pela variáveis do lado esquerdo de uma declaração de atribuição.

b) A programação não é procedural e sim baseadas em fatos, que podem ser associações entre coisas, e regras, que produzem fatos deduzidos a partir de outros.

c) Os programas consistem de substituições de parâmetros em funções que podem ser aplicadas sobre outras funções.

d) Objetos e classes são os blocos primitivos de construção de um sistema. Sistemas são vistos como coleções de objetos que se comunicam, enviando mensagens, colaborando para dar o comportamento global dos sistemas.

e) Versam sobre alterações de valores através de operações baseadas em atribuições e um fluxo de controle sequencial, condicional ou iterativo.

Plano de Aula

1 Código e nome da disciplina

ARA0066 PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

2 Semana/Tema

Semana 16: Tema - 6. PARADIGMAS: ESTRUTURADO, ORIENTADO A OBJETOS, FUNCIONAL E LÓGICO (ATIVIDADE PRÁTICA SUPERVISIONADA)

3 Objetivos

Apresentar o paradigma de programação lógico e as principais linguagens de programação que o utilizam.

4 Tópicos

6.3 PARADIGMA LÓGICO

5 Procedimentos de ensino-aprendizagem

Nesta aula, estaremos conectados com o conteúdo digital. O aluno explora e estuda, previamente, o conteúdo digital disponível em seu ambiente virtual. Durante a aula, este conteúdo será discutido em sala em atividade mediada pelo professor, detalhada abaixo.

Situação-problema:
Estamos em um momento da história na qual a Inteligência Artificial vem participando em diversos campos da nossa vida como por exemplo, no momento que interagimos com um atende virtual de um banco o qual precisa realizar processamento de linguagem natural para entender o que estamos perguntando e em seguida, a partir de uma base de conhecimento, deve nos orientar para a solução de nosso problema. Nesse ponto e levando em consideração que o sistema descrito pode ser considerado um Sistema Especialista que basicamente utiliza regras de inferência lógica do tipo SE - ENTÃO, como por exemplo:
SE saldo < valor_solicitado E limite = 0 ENTÃO "Seu saque não foi permitido pois o valor solicitado é superior ao seu saldo e você não possui limite no cheque especial"
Nesse caso os paradigmas que você estudou até agora são adequados para representar esse tipo de problema?

Metodologia:
Aula inicia com um rápido debate sobre os domínios de aplicação como do paradigma funcional como processamento de linguagem natural e sistemas especialistas. O professor deve apresentar ao aluno o software EXPERT SINTA disponível em https://iaexpert.academy/2016/09/13/ferramentas-para-ia-expert-sinta/ que implementa sistemas especialistas bastando que o aluno o alimente com uma base de conhecimento.

Atividade verificadora de aprendizagem:
O aluno deve utilizar as seguintes regras para criar um sistema especialista na ferramenta SINTA que verifique o problema de um motor.

Regra 1: se o motor recebe combustível e o motor tenta pegar então o problema é vela
Regra 2: se o motor não tenta pegar e as luzes não acendem então o problema é bateria e cabo
Regra 3: se o motor não tenta pegar e as luzes acendem então o problema é o motor de partida
Regra 4: se houver combustível no tanque e houver combustível no carburador então o motor recebe combustível
O professor deve apresentar a solução da atividade durante a aula.

6 Recursos didáticos

Laboratório de Informática com Internet com navegador Web instalado, equipado com quadro branco, projetor multimídia, acervo bibliográfico no ambiente virtual.

7 Leitura específica

[1] SEBESTA, Robert W. Conceitos de Linguagens de Programação. 11. edição. Porto Alegre: Bookman, 2018., Capítulo 12 (Suporte para programação orientada a objetos), páginas 726 a 757, Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9788582604694/

8 Aprenda +

[2] Vídeo "Sistemas Especialistas com Expert SINTA parte I", disponível em https://www.youtube.com/watch?v=bqzH8kRYmDY

Atividade Autônoma Aura:

Questão 1: (PaqTcPB- UEPB, 2012) No paradigma de programação lógico, um programa consiste basicamente de um conjunto de:

a) Métodos.

b) Regras.

c) Classes.

Questão 2 (CESPE/CEBRASPE-INMETRO, 2010) A respeito das características dos paradigmas e das linguagens de programação, assinale a opção correta.

a) As linguagens do paradigma de programação funcional, como o Prolog, não apresentam grandes restrições ao uso de estruturas de controle*(goto*), o que pode reduzir a legibilidade dos programas construídos sem limitações.

b) No paradigma lógico, que é suportado por linguagens de programação não imperativas, como o Lisp, os programas gerados são embasados em funções matemáticas.

c) O paradigma de programação orientado a objetos reúne linguagens, como o C++, que são declarativas, isto é, o foco está na especificação dos resultados desejados ao invés dos procedimentos

para produzi-los.

d) As linguagens procedimentais que dominaram o mercado antes da programação estruturada, tal como o COBOL, caracterizaram-se por utilizar amplamente os tipos abstratos de dados.

e) As linguagens imperativas, como o Pascal, são voltadas para a especificação da solução do problema, por meio do detalhamento do algoritmo e da especificação da ordem das instruções.

d) Aspectos.

e) Comandos.