TÉCNICAS DE PROGRAMAÇÃO - 1o Informática Matutino – Projeto I 2024

GALERIA VIRTUARTE

Este projeto tem por objetivo o desenvolvimento das habilidades dos estudantes na codificação de programas que processem arquivos texto, para organização de dados e emissão de relatórios e páginas web. Deve ser entregue, com seu relatório e pasta de programa com arquivos de códigos e arquivo de dados, até **25/04/2024.**

Uma galeria de arte virtual **registra obras que tem a exibir** em um arquivo texto cujas linhas possuem as informações dispostas como abaixo:

Arquivo de Obras

Ano da obra	- string[4]
Mês da obra	- string[2]
Estilo	- string[15]
Nome da obra	- string[20]
Autor da obra	- string[20]
Valor estimado	- real
urlFoto	- string[100]

Os primos do Montes Verdes – Jerci Maccari

Ano e mês da obra informam quando a obra foi finalizada. Estilo é um campo que informa a categoria da obra. Nome da obra descreve o nome dado à obra pelo seu autor. Autor da obra é o nome do autor da obra. Valor estimado é o preço dessa obra para venda de cópias impressas. O campo urlFoto indica o endereço de Internet onde uma miniatura (thumbnail) de imagem dessa obra pode ser encontrada.

Esses dados devem ser encapsulados como atributos públicos na classe Obra. Além desses dados, será encapsulada na classe um atributo lógico abertoParaGravacao que indicará se o arquivo foi aberto para gravação (true = **append**) ou para leitura (false = **read only**). Vemos o modelo da classe na figura abaixo, onde o símbolo + indica acesso público e # acesso protegido:

Obra
+AnoDaObra : string[4] +MesDaObra : string[2] +AutorDaObra : string[20] +NomeDaObra : string[20] +Estilo : string[15] +ValorEstimado : real +urlFoto: string[100] #abertoParaGravacao : bool #arquivo – arquivo aberto de acordo com nomeArq e paraGravacao
+Construtor __init__(nomeArq : string; paraGravacao : boolean) +lerCamposDoArquivo() +gravarCamposNoArquivo() +preencherCampos(novoAno, novoMes, novoAutor, novoNome, novoEstilo, novoValor, novaURL: string|) +fecharArquivo() +__str__() : string +compararCom(outraObra : Obra) : int

- **lerCamposDoArquivo** Se **abertoPara-Gravacao** valer **false**, será lida a **próxima linha** de dados do arquivo texto, armazenando seus dados nos atributos do objeto instanciado dessa classe, que representam os campos lidos do arquivo.
- **gravarCamposNoArquivo():** se o campo **abertoParaGravacao** valer **true**,.escreverá, no arquivo indicado pelo atributo **Arquivo**, os dados da obra encapsulada no objeto.
- **preencherCampos():** receberá como parâmetros os valores dos campos de dados da obra, os preencherá com espaços conforme necessário e os armazenará nos atributos de Obra que descrevem uma obra.
- **fecharArquivo()**: fecha o arquivo se este estiver aberto.
- **__str__() :** é uma função que concatena os atributos de dados da obra e **devolve a string** com essa concatenação, para leitura. Um exemplo de resultado seria:

2009 8 Bucólico Colheita de Uva IV Jerci Maccari 500.00 \imagens\colheitaUva4.jpg

Quando o programa de aplicação que usa a classe Obra precisar exibir os dados lidos, deverá chamar esse método função para receber a linha exibível com os dados. Também fica a cargo do programa de aplicação exibir um cabeçalho para esses dados quando eles forem exibidos.

- **compararCom()** recebe como parâmetro uma outra instância de Obra e a compara com a instância atual, através da concatenação dos atributos **AnoDaObra+MesDaObra+ AutorDaObra+NomeDaObra** de cada uma. Se essa concatenação da instância atual for menor que o da outra instância, o método devolverá o

valor -1. Se forem iguais, devolverá o valor 0. Se o apelido da instância atual for maior que o da outra instância, o método devolverá o valor 1.

Crie uma aplicação console em Python que exiba um seletor de opções repetitivo, com as opções abaixo:

1. Cadastro de obras de arte
2. Listagem de obras de arte
3. Página web de obras de arte
4. Triangulo de Pascal
0. Terminar a execução do programa

Após ler a opção escolhida pelo usuário, se esta for entre 1 e 4, o programa deverá chamar a função específica que trata da operação associada à opção.

O módulo principal exibirá as opções citadas no início deste enunciado de forma repetitiva, parando a execução quando a opção 0 for digitada pelo usuário.

1. Cadastro de Obras em um arquivo de Obras

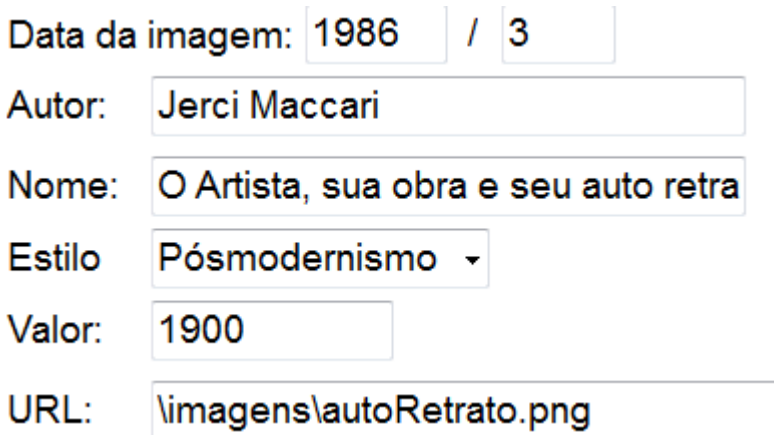
Quando o usuário selecionar a opção 1 - Cadastro, deve-se solicitar, através do OpenDialog do módulo TKinter, o nome do arquivo texto de obras. Em seguida, deverá instanciar um objeto da classe Obra, informando o nome do arquivo escolhido com abertura para **gravação após seu final**.

Em seguida, deve-se solicitar ao usuário que digite cada um dos campos de dados da obra, armazenar os dados digitados no objeto Obra acima instanciado, por meio do método preencherCampos() e gravar um registro de obra ao final do arquivo aberto, com o uso do método gravarCamposDoArquivo().

Após isso, deve-se apagar a tela e retornar-se à digitação de um novo registro.

Quando o usuário digitar um ano igual a 0, esta opção deve ser finalizada e o fluxo de execução deve fechar o arquivo e retornar ao seletor de opções repetitivo.

Ao lado temos um esboço de como a tela console poderia ficar (sem caixa de texto gráfico, obviamente):



O artista, sua obra e o autoretrato - Jerci Maccari

2. Listagem de Obras

Nessa opção, o programa deverá solicitar o nome do arquivo de obras com o OpenDialog do TKinter e instanciar um objeto da classe Obra, abrindo o arquivo para leitura. Em seguida, usando o método lerCamposDoArquivo(), leia o arquivo sequencialmente e exiba os seus registros na tela, usando como modelo de layout a figura abaixo. Calcule e exiba os totais.

Ano	Mês	Est	Nome da Obra	Autor	Valor	URL
1495	12	Medieval	Jardim das Delicias Terrenas	H. Bosch	2300.00	\imagens\JardimDeliciasTerrenas.jpg
1512	10	Medieval	Sete Pecados Capitais	H. Bosch	2300.00	\imagens\SetePecadosCapitais.jpg
1928	5	Moderno	Abaporú	Tarsila do Amaral	5000.00	\imagens\abaporu.jpg
1928	6	Moderno	Paisagem de Petrópolis	Candido Portinari	2800.00	\imagens\paisagemDePetropolis.png
1928	8	Moderno	Antropofagia	Tarsila do Amaral	5000.00	\imagens\abaporu.jpg
1952	1	Moderno	Caranguejo, Caju e Garrafas	José Pancetti	1920.00	\imagens\caranguejoCajuGarrafa.
1986	3	Pósmoder	O Artista, sua obra e seu au	Jerci Maccari	1900.00	\imagens\autoRetrato.png
2009	5	Moderno	Colhendo Uva II	Jerci Maccari	3500.00	\imagens\colheitaUva2.jpg
2009	8	Moderno	Colheita de Uva IV	Jerci Maccari	2000.00	\imagens\colheitaUva4.jpg
2013	1	Moderno	Santa Ceia	Jerci Maccari	1500.00	\imagens\santaCeia.jpg
2013	7	Moderno	Os Primos de Montes Verdes	Jerci Maccari	2500.00	\imagens\PrimosMontesVerdes.jpg
2013	9	Moderno	Safra de Trigo	Jerci Maccari	4500.00	\imagens\SafraDeTrigo3.jpg
			Número de obras : 12		Valor:	35220.00

Quando se terminar a listagem, deve-se fechar o arquivo e o fluxo retornar ao seletor de opções repetitivo.

3. Relatório HTML

Essa operação será disparada quando o usuário selecionar a opção 3 - Relatório do seletor de.

A função que trata essa operação deverá solicitar com o OpenDialog doTKinter o nome de um arquivo de obras. Usando um objeto da classe Obra, abrimos o arquivo escolhido, o lemos e geramos um relatório como uma tabela em um arquivo HTML (obras.html) que deverá ser posteriormente mostrado através de um browser. Pesquise como fazer para que um navegador, como o Chrome, possa ser chamado a partir de seu programa já com o arquivo gerado aberto..

Na página seguinte temos um modelo de como essa página poderá ser. Você pode usar quaisquer tags e recursos aprendidos na disciplina **TI123 - Desenvolvimento para Internet**.

Observe que existe uma quebra de nível no relatório, quando mudam-se os dados de ano da obra.

4. O Triângulo de Pascal é um triângulo numérico infinito formado por um arranjo de valores $\binom{n}{k}$, onde n representa o número da linha (posição vertical) e k representa o número da coluna (posição horizontal). O triângulo foi descoberto pelo matemático chinês Yang Hui, e 500 anos depois várias de suas propriedades foram estudadas pelo francês Blaise Pascal.

Por exemplo, para 8 linhas temos:

```
1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
1   5   10  10  5   1
1   6   15  20  15  6   1
1   7   21  35  35  21  7   1
```

	0	**1**	**2**	**3**	**4**	**5**
0	1					
1	1	1				
2	1	2	1			
3	1	3	3	1		
4	1	4	$\overline{6}$	$\overline{4}$	1	
5	1	5	10	$\underline{\overline{10}}$	5	1

Colunas k

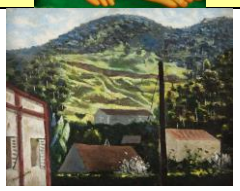
Linhas n

Outro exemplo: Triângulo de Pascal para 6 linhas

O valor de $\binom{n}{k}$ é dado pela fórmula **fatorial(n) / (fatorial(k) * fatorial(n-k))**

Codifique uma classe Matematica que tenha como atributo inteiro a variável _numeroBase,cujo valor deverá ser estabelecido pelo programa, como um parâmetro ao instanciar um objeto dessa classe na opção 4 do móduço principal. Nessa classe, codifique:

- método fatorial (x :int) -> int que calcula e retorna o fatorial do parâmetro x. O fatorial de um valor x é igual ao produtório de todos os inteiros positivos entre 2 e x
- método triangulo_de_Pascal(), que use o valor do atributo _numeroBase da classe como sendo o número de linhas L de um Triângulo de Pascal e retorne uma lista de strings onde cada posição contenha uma linha do Triângulo correspondente a esse número de linhas. Note que as linhas (n) variam de 0 a L-1, enquanto, **para cada** valor de linha (n), as colunas (k) variam de 0 a n. Para cada valor de linha e coluna, deve-se calcular o fatorial de n, o fatorial de k e o fatorial de n-k, usando-os na fórmula acima para determinar o valor que será exibido na tela. Cada linha, portanto, será uma string que conterá os números previstos para essa linha, e será, depois de gerada, adicionada à lista de strings que será retornada pelo método triangulo_de_Pascal(). Lembre-se de formatar os valores de cada string/linha do triângulo com tamanho de, no mínimo, 6 posições. Lembre-se que a classe Matematica possuirá o método fatorial(), que deverá ser chamado no método triangulo_de_Pascal() para calcular os fatoriais dos valores n, k e n-k.

RELATÓRIO DE OBRAS DA GALERIA VIRTUAL					
Ano/Mes	**Dados**	**Estilo**	**Autor**	**Valor**	**Imagem**
1512 / 1	Jardim Das Delícias Terrenas	Medieval	H. Bosch	2300.00	
1512 / 10	Sete Pecados Capitais	Medieval	H. Bosch	2300.00	
Total				**4600.00**	
1928 / 5	Abaporú	Moderno	Tarsila do Amaral	5000.00	
1928 / 6	Paisagem de Petrópolis	Moderno	Cândido Portinari	2800.00	
1928 / 8	Antropofagia	Moderno	Tarsila do Amaral	5000.00	
Total				**12800.00**	
...	...	...	...	...	...
2013 / 7	Os Primos de Montes Verdes	Moderno	Jerci Maccari	2500.00	
2013 / 9	Safra de Trigo III	Moderno	Jerci Maccari	500.00	
Total				**8500.00**	
Total Geral				**35220.00**	

Paisagem Rural V – Jerci Maccari

Gratidão especial ao grande artista Jerci Maccari, que generosamente e gratuitamente cedeu as imagens de suas obras para ilustrar este projeto.

Saiba mais sobre os artistas citados:

http://jercimaccari.com.br/

http://enciclopedia.itaucultural.org.br/pessoa1334/jose-pancetti

http://tarsiladoamaral.com.br/

http://www.laparola.com.br/hieronymus-bosch-o-misterio-por-tras-de-sua-obra

http://www.portinari.org.br/

O trabalho deverá ser realizado **em dupla** e o relatório deve descrever as atividades desenvolvidas, o auxílio da monitoria, erros e problemas encontrados, suas soluções, o tempo gasto em cada atividade, bem como a conclusão sobre o projeto e sugestões. Imagens ilustrativas de erros, suas soluções e da execução do programa são muito importantes também.

Entregar pasta do projeto, compactada, seguindo o padrão de nomenclatura de pasta **ra1**_**ra2**_Proj1, sendo ra1 o menor dos RAs dos estudantes da dupla autora, e ra2 o maior dos RAs. Exemplo: **24105**_**24156**_Proj1.rar.

Nessa pasta devem constar o relatório de desenvolvimento em **formato PDF**, os arquivos fontes Python e um arquivo de dados para testes, gerado pelo seu próprio programa.

Entrega: 25/04/2022.