



Universidade Federal de Goiás
Instituto de Informática

POO – Laboratório (22/10/2024): Padrões de Projeto (*Design Pattern*)
Estudo de Caso: *Abstract Factory*

(Em grupo, o mesmo do Trabalho Final da Disciplina (TFD))

Submeta as respostas do documento em um único arquivo (.doc, .odt. ou .pdf) submetido em grupo, o grupo do TFD. Somente um estudante do grupo deverá submeter esta Tarefa. Coloque como comentário ao enviar no SIGAA no documento matrícula, nome dos participantes do grupo, turma. Lembro que no arquivo enviado deve ter as identificações dos alunos (matrícula e Nome), do prof. e da disciplina. Embora somente um estudante do grupo deverá enviar a resposta é muito importante cada estudante do grupo ter uma cópia, pois a segunda avaliação será no lab.

Estudante(s) do Grupo matrícula(s) e Nome(s):

1. _____
2. _____
3. _____
4. _____

1) Responda as seguintes questões sobre o código-fonte AbstractFactory_java.zip disponibilizado na página da disciplina no SIGAA.

Obs.: Antes de responder à questão:

- Crie um Projeto Java no Eclipse
- Copie o arquivo compactado AbstractFactory_java.zip e descompacte os arquivos no mesmo.

- a) Qual é a classe .java que deve ser executada para gerar um *output*? Por que?
- b) Faça uma pequena modificação no fonte da classe da letra “a” inserindo e imprimindo a(s) matrícula(s) e o nome(s) do grupo. Capture a tela (<Print Screen>) da janela de execução ao rodar o output (saída) na linguagem Java.
- c) Qual é o arquivo que contém o Diagrama de Classe no UMLet da Abstract Factory? Capture a tela (<Print Screen>) com a figura do Diagrama de Classe do UMLet.

- 2) Responda as seguintes questões sobre o código-fonte AbstractFactory_Python.zip disponibilizado na página da disciplina no SIGAA.

Obs.: Antes de responder à questão:

- Instale a extensão do Python da própria Microsoft no Visual Studio Code, caso já não esteja instalada.
- Crie um diretório para o projeto Python
- Copie o arquivo compactado AbstractFactory_Python.zip e descompacte os arquivos no mesmo.

- a) Qual é a classe .py que deve ser executada para gerar um *output*? Por que?
- b) Faça uma pequena modificação no fonte da classe da letra “a” inserindo e imprimindo a(s) matrícula(s) e o nome(s) do grupo. Capture a tela (<Print Screen>) da janela de execução ao rodar o output (saída) na linguagem Python.
- c) Copie o Diagrama de Classe do UMLet da letra “c” do exercício 1. Modifique o Diagrama de Classe (use as propriedades de classes necessárias com a paleta do UMLet) de modo a ele se tornar compatível com o desenho das classes do arquivo AbstractFactory_Python.zip. Copie o arquivo do UMLet, renomeie para inserir no nome + “_Python.uxf). Coloque a(s) matrícula(s) e nome(s) do(s) estudante(s) do grupo como note no Diagrama de Classe. Salve o arquivo .uxf e envie dentro do arquivo compactado (zipado) com a solução do grupo para esta letra da questão.
- d) Como é implementado o conceito de interface usando a linguagem Python?

- 3) Responda as seguintes questões sobre o código-fonte AbstractFactory.ipynb disponibilizado na página da disciplina no SIGAA.

Obs.: Antes de responder à questão:

- Abra o Google Colab usando a conta do e-mail (@ufg.br ou @gmail.com).
Site: <https://colab.research.google.com/notebooks/intro.ipynb>
- Faça o upload do arquivo AbstractFactory.ipynb, opção, **Arquivo -> Fazer Upload de notebook.**

- a) O que é o Google Colab? Qual a sua utilidade para rodar código em Python?
- b) O que é uma célula de código e uma célula de texto no Google Colab? Qual é a utilidade de usá-la e quando devo usar uma ou outra?
- c) Como é possível executar o código .ipynb, neste arquivo qual é a célula Qual(is) é ou são a(s) célula(s) que deve(m) ser executada para gerar um *output*? Qual é a célula que tem a classe que permite rodar o programa?

- d) Ao abrir o arquivo AbstractFactory.ipynb em um editor de texto puro como no bloco de notas do Windows ou outro software que abre .txt puro. Que tipo de código fonte é encontrado ao abrir este arquivo desta forma? Existe alguma relação com o padrão *markdown*? O *Google Colab* consegue ler arquivos com *tags markdown*?

Fonte básico de informações sobre o tema:

https://colab.research.google.com/notebooks/markdown_guide.ipynb

- e) Faça uma pequena modificação no fonte AbstractFactory.ipynb inserindo e imprimindo a(s) matrícula(s) e o nome(s) do grupo. Salve as mudanças e modifique e renomeie o arquivo .ipynb. renomeie para inserir no nome + “_modificado.ipynb). Coloque a(s) matrícula(s) e nome(s) do(s) estudante(s) como uma célula de texto inicial do código fonte. Salve o arquivo .ipynb e envie dentro do arquivo compactado (zipado) com a solução do grupo para esta letra da questão.
- f) Além da extensão do nome do arquivo no formato .py da questão 2 e da extensão no formato .ipynb. Qual ou quais são as diferenças nos códigos-fontes de ambos que permitem rodá-los sem erros cada um no seu ambiente?
- g) Ao analisar o *output* dos três exercícios responda se existe alguma diferença na lógica de programação ou as três soluções se equivalem logicamente? (Justifique a sua resposta)

- 4) Responda as seguintes questões em relação ao *Design Patterns: Abstract Factory*.

- a) O que é um *Design Patterns*?
- b) Em que contexto pode-se ser útil usar uma solução usando *Design Patterns* do tipo *Abstract Factory*? Exemplifique sem usar o exemplo do lab. da Fábrica de Pizza ou Pizzaria.
- c) Quais os principais prós e contras de uma solução envolvendo o padrão *Abstract Factory*?
- d) Quais são as relações deste padrão *Abstract Factory* com outros padrões de projeto?

Obs.: Existem várias informações sobre Padrões de Projetos na internet. Boas fontes básicas de informações sobre este padrão para responder estas questões podem ser encontradas no *link* abaixo:

<https://refactoring.guru/pt-br/design-patterns/abstract-factory>