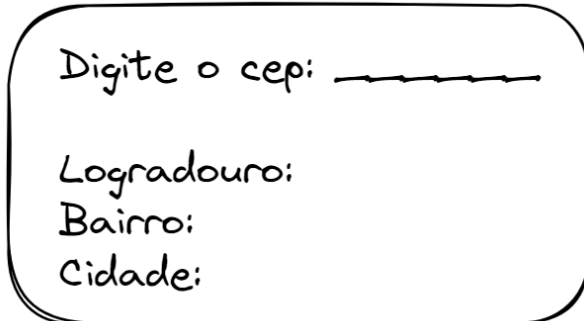


Exercícios e Correções DDD - FIAP 1TDSR

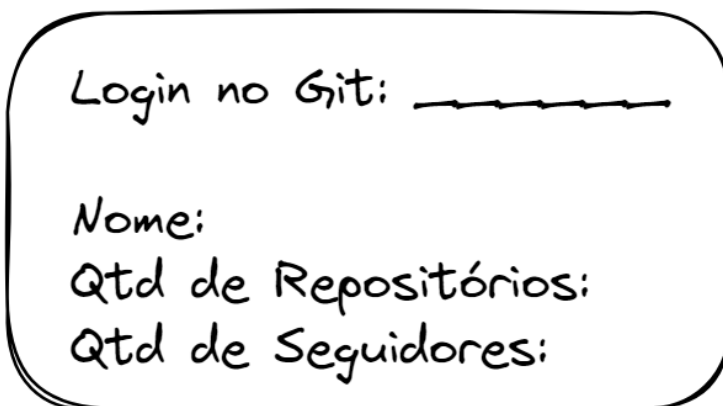
Crie um programa em Java que permita consultar endereços através de um Cep. O programa deve consumir a API do ViaCep (**Exemplo: viacep.com.br/ws/01001000/json/**), permitindo o usuário digitar o seu cep e obter as informações abaixo:



A hand-drawn form with a rounded rectangle border. It contains the following text: "Digite o cep:" followed by a line of ten dashes for input. Below this, it lists "Logradouro:", "Bairro:", and "Cidade:" on separate lines.

Correção: https://drive.google.com/drive/u/1/folders/1LtelMiaxkcDpVfsXZ4qphCT0TFisCJ_8

Crie um programa em Java que permita consultar Devs no GitHub. O programa deve consumir a API do GitHub (**Exemplo: <https://api.github.com/users/joseffe10>**), permitindo o usuário digitar algum login do GitHub e após isso retornar as informações abaixo:



A hand-drawn form with a rounded rectangle border. It contains the following text: "Login no Git:" followed by a line of ten dashes for input. Below this, it lists "Nome:", "Qtd de Repositórios:", and "Qtd de Seguidores:" on separate lines.

Correção:

<https://drive.google.com/drive/u/1/folders/19A7kcZsbaPNLEgvqEYUsbUmfSCXcE6SC>

Crie um programa em Java que permita converter o valor de R\$ para outras moedas. O programa deve consumir a API de Conversão (**Documentação: <https://docs.awesomeapi.com.br/api-de-moedas>**), permitindo o usuário digitar o valor em reais e escolher qual o destino da conversão:

Exemplo de uso (o atributo **ask** é o que possui o valor da conversão):

Real para Dólar: <https://economia.awesomeapi.com.br/json/last/USD-BRL>

Real para Euro: <https://economia.awesomeapi.com.br/json/last/EUR-BRL>

Real para Bitcoin: <https://economia.awesomeapi.com.br/json/last/BTC-BRL>

Conversor de Moedas

Qual o valor em R\$: _____

Deseja converter para:

1. Dólar
2. Euro
3. Bitcoin

Opção escolhida: _____

Resultado: _____

Correção:

<https://drive.google.com/drive/u/1/folders/1t9furMNQWx1pGg1Akj63tPsJPciZuoS3>

Crie um programa em Java que permita cadastrar pessoas e seu respectivo endereço. Ao cadastrar o endereço deixe a pessoa, digitar apenas o cep e o restante dos campos, deve ser completado de acordo com o retorno da API, como rua, bairro, cidade e estado. A pessoa, só deve cadastrar seu nome, email, número da casa e complemento. Todo esse conteúdo deve ser cadastrado em um ArrayList. O programa deve permitir, Incluir, Alterar, Excluir e Consultar Pessoas.

Correção:

<https://drive.google.com/drive/u/1/folders/1mDA-PKMWRwspRkWMB0wVWOqp6KqkVJsW>

Crie um programa para testar se o nome de um domínio na Web está disponível para criação de um site. Exemplo: joseffe.com.br

<https://brasilapi.com.br/api/registrobr/v1/joseffe.com.br>

Se o campo "status" retornar AVAILABLE, aparecer na tela Domínio disponível, caso contrário, aparecer Domínio já utilizado e exibir a data de expiração.

Correção: https://drive.google.com/drive/u/1/folders/12ldvYLsulyXaRaV5v-Juezd3-_aJNHJr

Crie um programa para identificar feriados a partir de uma data digitada. O programa deverá transformar a data digitada no formato dd/mm para o formato 2023-mm-dd. Sendo assim, deverá identificar no retorno da API <https://brasilapi.com.br/api/feriados/v1/2023> se a data

digitada é um feriado ou não. Se for feriado, retornar o nome do feriado, se não for feriado, retornar os feriados do mês digitado.

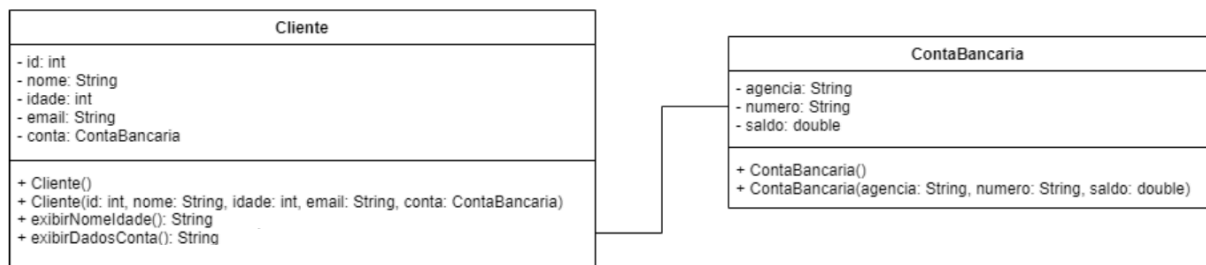
Correção sem HashMap:

<https://drive.google.com/drive/u/1/folders/1uHELlhh1-2zCZj5iZE2dqcfjUlo9AfKtR>

Correção com HashMap:

<https://drive.google.com/drive/u/1/folders/1DhLHXZaAGOPMhuam753hChoXLS2pwoMp>

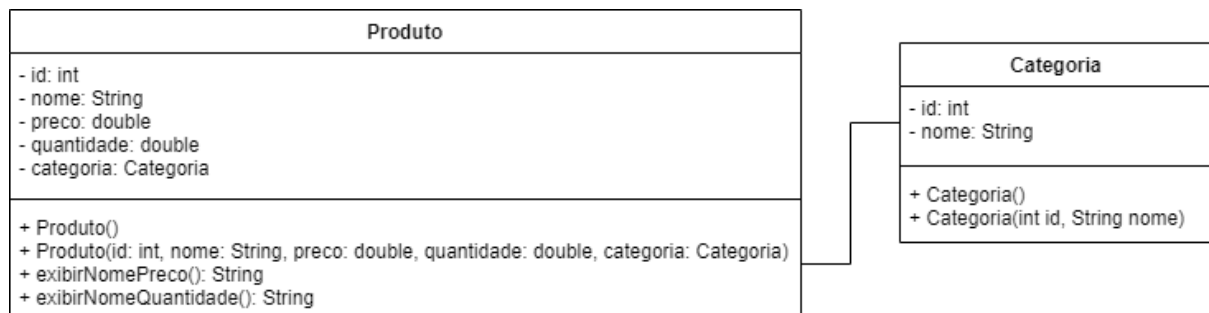
Utilizando **HashMap**, crie um sistema com as classes conforme o Diagrama de Classe (UML) abaixo. Crie Getters e Setters para todos os atributos das classes. Crie um programa que utilize essas classes para cadastrar clientes e pergunte para cada cliente se ele tem ou não conta bancária. Caso o cliente tenha, permita ele cadastrar os dados da conta bancária. Ao final, exibir todos os clientes e suas respectivas contas bancárias, se houver.



Correção:

https://drive.google.com/drive/u/1/folders/1EGJ1AjSZGRPfwWPMCyj22b2HJruMzr_

Utilizando **HashMap**, crie um sistema com as classes conforme o Diagrama de Classe (UML) abaixo. Crie Getters e Setters para todos os atributos das classes. Crie um programa que utilize essas classes para cadastrar produtos e pergunte para cada produto se ele tem ou não uma categoria. Caso o produto tenha, permita ele cadastrar os dados da categoria. Ao final, exibir todos os produtos e suas respectivas categorias, se houver.



Correção:

<https://drive.google.com/drive/u/1/folders/1Fu3hmJhBieq1WjFOWAB3LMCbPVBYA0JV>

Vamos fazer um sistema de cadastro de clientes e sua respectiva conta bancária. O sistema deve permitir, inclusão de cliente, depósito na conta, saque na conta, exclusão de cliente e consulta de cliente e extrato.

O cliente deve escolher entre conta corrente e conta poupança no momento de se cadastrar. Utilize HashMap para armazenar os clientes e contas. Utilize Herança e Polimorfismo nos métodos da classe das contas.

Correção:

https://drive.google.com/drive/u/1/folders/1N3EKqdT_FdDEhXIOq_BZ2ESldhT3dIW

Vamos fazer um sistema de cadastro de contas. O sistema deve ter as opções Conta Corrente, Conta Poupança e Conta Salário. Implemente o conceito de Classe Abstrata nesse sistema. Faça o Diagrama de Classe (UML) desse sistema também.

Correção:

https://drive.google.com/drive/u/1/folders/1BZTQi-r_KVnK0IzYbTH36ZMYCsRJWZe_

Vamos fazer um sistema de cadastro de pessoas. O sistema deve ter as opções Pessoa Física e Pessoa Jurídica. Implemente o conceito de Classe Abstrata nesse sistema. Faça o Diagrama de Classe (UML) desse sistema também.

Correção:

<https://drive.google.com/drive/u/1/folders/1M6YNxOAsXITVqZzmycGjcVLAQR05ZsdC>

Vamos fazer um sistema de cadastro de contas. O sistema deve ter as opções Conta Corrente, Conta Poupança e Conta Salário. Implemente o conceito de Herança e Interface nesse sistema. Faça o Diagrama de Classe (UML) desse sistema também.

Correção:

https://drive.google.com/drive/u/1/folders/1LmEXWtZsuBOBy3E9tU3_YxYy_g0CU9jY

Vamos fazer um sistema de cadastro de pessoas. O sistema deve ter as opções Pessoa Física e Pessoa Jurídica. Implemente o conceito de Herança e Interface nesse sistema. Faça o Diagrama de Classe (UML) desse sistema também.

Correção:

<https://drive.google.com/drive/u/1/folders/1Q1cYFFzgS-s8SQ6byyouHk0PMYB2mpLh>

Sistema de Cadastro de Tecnologias (back e front):

- Uma interface para representar qualquer tecnologia de desenvolvimento, definindo métodos para Descrição da Linguagem e Descrição do Framework;

- Uma classe abstrata para representar as tecnologias de desenvolvimento. Seus atributos devem ser linguagem de programação, principal framework, descrição da linguagem e descrição do framework;
- Classes para representar backend e frontend. A primeira deve receber a informação se é possível containerizar o desenv e a segunda deve receber a informação se é compatível com web ou mobile. Essas classes devem herdar da classe abstrata e implementar a interface;
- No programa principal, pergunte ao usuário quantas tecnologias ele deseja criar. Em seguida, para cada tecnologia, pergunte se deseja criar uma tecnologia de backend ou de frontend, solicitando os dados necessários para criá-las. Todas as tecnologias criadas devem ser armazenadas em um ArrayList/HashMap próprio. Ao final, após cadastrar as tecnologias, finalmente, imprima: (a) os dados (linguagem e principal framework); (b) as descrições;
- Desenvolva o tratamento de exceções para o programa desenvolvido;

Correção:

https://drive.google.com/drive/u/1/folders/1TgvAvly1z6h6XRyEGHg5J_QW9n3uMcrW

Crie um programa que tenha uma variável String chamada **alunosVestibular**. Essa **variável** deve ter o conteúdo abaixo:

String alunosVestibular = Jose dos Santos,7,Sao Paulo;Sandra Silva,6.5,Sao Jose do Rio Preto;Augusto Soares,8,Sao Paulo;Vanderlei Azevedo,5.65,Santos;Vanessa Ferreira,9,Sao Paulo;Natan Cruz,10,Sao Paulo.

Essa variável contém a lista das pessoas que fizeram o vestibular extra que a faculdade ofereceu. Perceba que a variável possui 6 alunos, todos eles separados por ; (ponto e vírgula).

Nesse conteúdo temos o nome do candidato, nota da prova e a sua cidade de origem.

Com isso, criar um programa onde irá ler esse conteúdo, criar um objeto para cada linha do conteúdo e incluir todos em um HashMap. Apenas incluir no HashMap, os candidatos que tiraram nota igual ou superior a 7.

Ao final, exibir todos os candidatos incluídos no HashMap, nesse formato:

Nome:
Nota:
Cidade:

Nome:
Nota:

Cidade:

Correção:

https://drive.google.com/drive/u/1/folders/14YJT9Ou3HEWiMcP4k2q-_ssrFOum2yh9

Crie um programa que tenha uma variável String chamada **baseDados**. Essa **variável** deve ter o conteúdo abaixo:

String baseDados = CJose dos Santos,42,Sao Paulo;CSandra Silva,36,Sao Jose do Rio Preto;CAugusto Soares,22,Sao Paulo;CVanderlei Azevedo,45,Santos;CVanessa Ferreira,27,Sao Paulo;PMouse,1,9.90;PTeclado,3,19.90;PMonitor,2,349.90;PHD SSD,2,199.90;PProcessador,1,350.00

Essa variável está reunindo informações de clientes (nome, idade e cidade) e produtos (nome, quantidade em estoque e preço). Perceba que os clientes começam com "C" e os produtos começam com "P". Além disso, observe também que todos eles são separados por ; (ponto e vírgula).

Criar uma classe Cliente e Produto com atributos baseados no que temos na variável String: clientes (nome, idade e cidade) e produtos (nome, quantidade em estoque e preço).

Com isso, criar um programa Java onde irá ler esse conteúdo, criar um objeto para cada registro e incluir em um ArrayList em sua respectiva classe.

Ao final, exibir todos as informações nesse formato:

Clientes:

Nome:

Idade:

Cidade:

Produtos:

Nome:

Qtd em estoque:

Preço:

Correção:

https://drive.google.com/drive/u/1/folders/1kt_7D0_OJ3CCAGHqiBNLY8lsgokrDMA-

Crie um programa que solicite ao usuário uma lista de compras. O programa deve solicitar 10 produtos, contendo nome, quantidade e preço unitário. Todas essas informações devem ser gravadas em um arquivo chamado lista.txt, separadas por vírgula.

Exemplo:

Arroz,1,21.90
Feijao,3,8.50
Pizza,2,14.90

Criar tratamentos e validações no programa.

Correção:

https://drive.google.com/drive/u/1/folders/1qdRhmaB7shtlF_KG0T1RNkVatBPpo56h

A partir do programa acima, crie um programa que leia o arquivo lista.txt e calcule o preço total de cada item e escreva em um novo arquivo chamado total.txt.

Exemplo:

Arquivo: lista.txt

Arroz,1,21.90
Feijao,3,8.50
Pizza,2,14.90

O arquivo total.txt deverá ter:

Arroz,21.90
Feijao,25.50
Pizza,29.80

Ambos os arquivos devem estar na mesma pasta.

Correção:

<https://drive.google.com/drive/u/1/folders/1aR9WFaeZwmZAOhpxsig8o9poQdxjREBH>

Crie um cadastro de alunos, contendo informações de id, nome, ra e curso. Esse programa deve ter o seguinte menu:

- 1 - Incluir alunos
- 2 - Atualizar alunos
- 3 - Excluir alunos
- 4 - Listar alunos
- 5 - Listar um aluno

Todos os alunos cadastrados devem ser gravados no banco de dados **Oracle**.

Correção: