UDESC – CCT
Bacharelado em Ciência da Computação
Exame POO - 2021.1

Professor: Fabiano Baldo

01 de setembro de 2021

Questão 1 (3,5 pontos)

1. **(3,0)** Dadas as classes do diagrama da Figura 1, crie uma classe chamada **Gastos** e implemente a função **public exibirGastos(List<ITaxavel> itens)**. Tal função itera sobre um List de ITaxavel, exibindo os valores dos produtos, dos impostos dos produtos e dos seus tipos (Comida, Veiculo ou Filme). O imposto da Comida é 15% de seu preço, do veículo é 250 reais, caso ele seja anterior a 2000, ou 450 reais, caso contrário. Já para os Filmes, o valor do imposto leva em consideração a duração (em minutos) dele. O valor corresponde a 1,5% da duração total do filme.

2. **(0,5)** Crie um método **main()** e instancie uma lista contendo ao menos 2 ITaxavel de cada uma das classes. Chame a função **exibirGastos()** para a lista instanciada e imprima as informações sobre os gastos na tela.

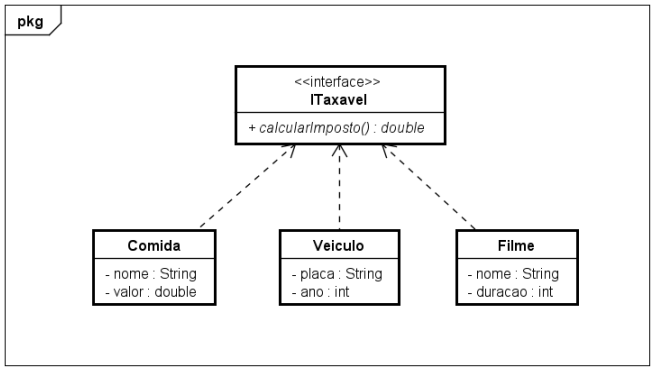

Figure 1: Diagrama UML

Questão 2 (1,5 pontos)

Apresente os resultados da execução dos algoritmos 1 e 2 para os parâmetros a seguir:

a) (0,5) A = 9, B = 10 e C = 11;

b) (0,5) A = 10, B = 11 e C = 12;

c) (0,5) A = 11, B = 12 e C = 13;

Caso deseje testar o algoritmo, a saída para a entrada $A = 3$, $B = 4$ e $C = 5$ é igual a -73.

Algorithm 1: Função main()
Atenção: A chamada **imprima** NÃO pula uma linha ao final!

```
   Data: Valores inteiros A,B e C
 1 Tente executar:
 2     calculaX(A);
 3     calculaX(B);
 4 Capture qualquer exceção e coloque-a em E:
 5     imprima (Mensagem em E);
 6 Fim tente;
 7 Tente executar:
 8     calculaX(C);
 9 Capture qualquer exceção e coloque-a em E:
10     imprima (Mensagem em E);
11 Fim tente;
```

Algorithm 2: Função calculaX()

```
   Data: Valor inteiro X
 1 Se X ∈ Primos:
 2   Tente executar:
 3     Lance uma exceção contendo o valor de x * x em sua mensagem;
 4   Capture qualquer exceção e coloque-a em E:
 5     X ← X - 2;
 6     Lance uma exceção contendo o valor de x em sua mensagem;
 7   Fim tente;
 8 Senão:
 9   Lance uma exceção contendo o valor de x − 10 em sua mensagem;
10 Fimse;
```

Questão 3 (1,5 pontos)

A partir das classes descritas na Figura 2, compare os três trechos de código e informe para qual deles a saída será igual a 29. Para os demais informe o porquê da saída não ser o valor esperado.

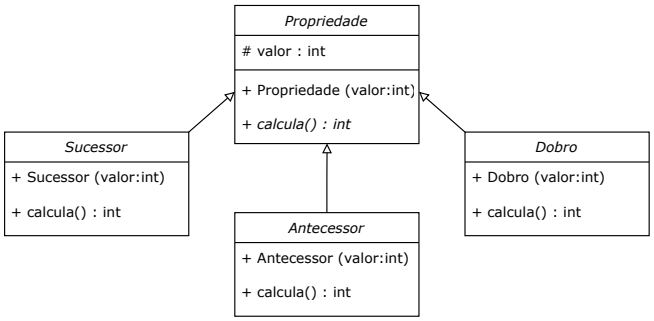

Figure 2: Classe Abstrata Propriedade e suas classes filhas Sucessor, Antecessor, Dobro e Triplo. O método abstrato **calcula()** deve ser implementado pelas classes filhas, retornando o valor ao aplicar a propriedade que dá nome a classe.

```
Sucessor s = new Sucessor(3);
Antecessor a = new Antecessor(4);
Dobro d = new Dobro(5);

Propriedade[] props = { s, a, d };

int soma = 0;

for (Propriedade prop : props) {
    soma += prop.calcula();
}

System.out.println(soma);
```

Código 1: Soma 1.

```
Propriedade s = new Sucessor(6);
Propriedade a = new Antecessor(7);
Propriedade d = new Dobro(8);

Propriedade[] props = { s, a, d };

int soma = 0;

for (Propriedade prop : props) {
    soma += prop.calcula();
}

System.out.println(soma);
```

Código 2: Soma 2.

```
Sucessor s = new Propriedade(6);
Antecessor a = new Propriedade(7);
Dobro d = new Propriedade(8);

Propriedade[] props = { s, a, d };

int soma = 0;

for (Propriedade prop : props) {
    soma += prop.calcula();
}

System.out.println(soma);
```

Código 3: Soma 3.

Questão 4 (3,5 pontos)

1. **(3,0)**Implemente as classes Cor e PaletaCores (ver Figura 3). A classe Cor possui quatro atributos: as coordenadas R,G e B (valores inteiros que variam de 0 a 255) e um nome. A classe Cor também deve sobreescrever o método **equals()**, comparando apenas as coordenadas R,G e B. Já a classe PaletaCores, possui uma lista de cores, bem como um método **adicionarCor()** e um método **filtrar()**. O método **adicionarCor()** deve receber um objeto Cor e adicionar a lista de cores da paleta caso a cor não se encontre na lista. Se a cor já estiver na lista (implemente e utilize o método **equals()** para verificar isso), uma Exception deve ser lançada com uma mensagem informando que a

cor já existe na paleta. Por fim, o método **filtrar()** irá retornar apenas as cores que não possuem tons de azul, isto é, cores cuja coordenada B vale 0.

2. **(0,5)**Crie um método **main()** e instancie uma paleta e as cores da tabela 1, adicionando-as a paleta e tratando as exceções corretamente.

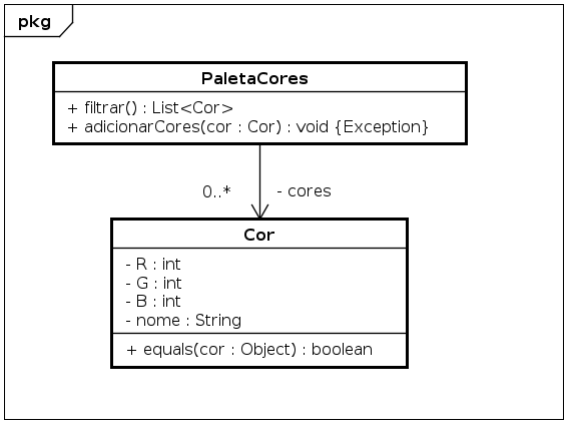

Figure 3: Classes Cor e PaletaCores.

Nome	R	G	B
Azul	0	0	255
Vermelho	255	0	0
Amarelo	255	255	0
Magenta	255	0	255
Branco	255	255	255

Table 1: Tabela de cores.