



Nome: Rômulo Magno Rodrigues Borio

Turma: 24

Matéria: CES-22

Professor: Edgar Yano

Laboratório 2

Para realização do problema proposto, foi realizado o diagrama de classes exposto abaixo.

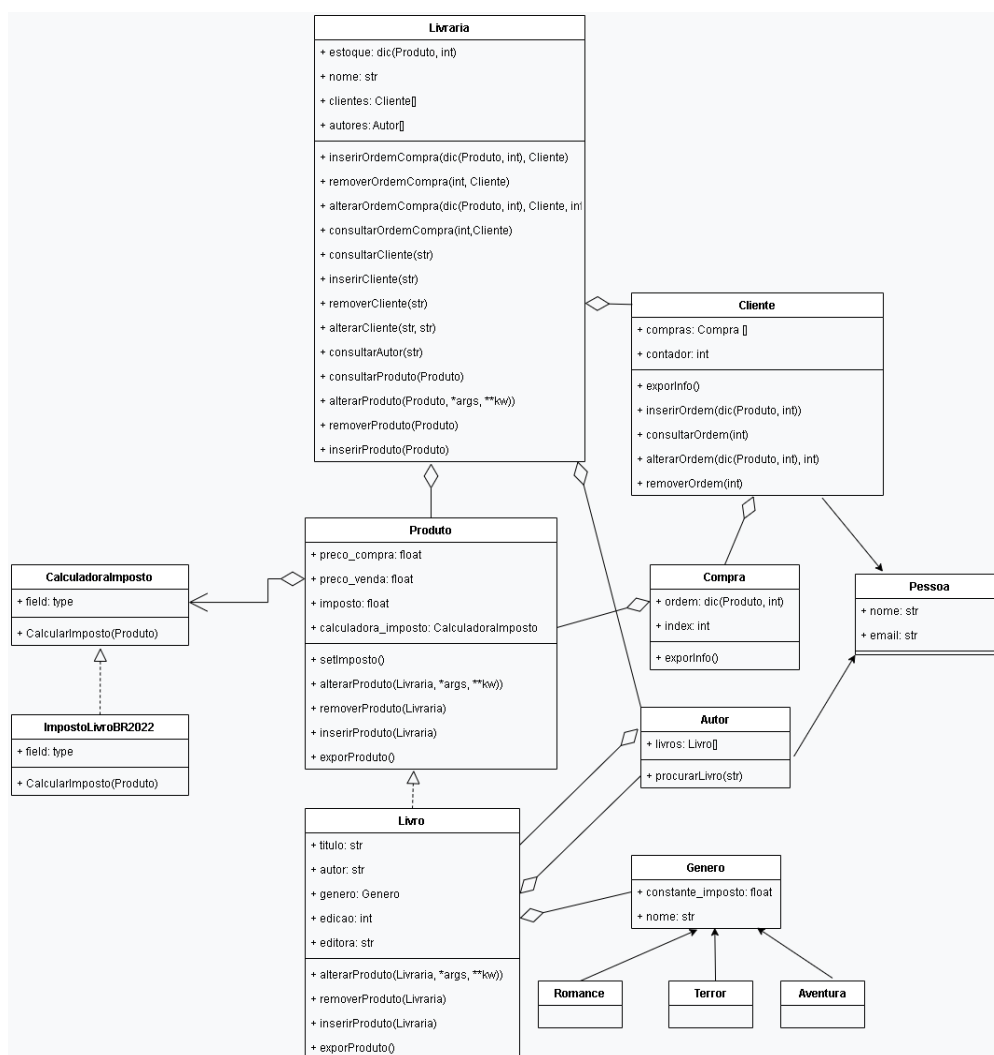


Figura 1: Diagrama de classes do problema proposto

Durante sua realização, algumas decisões de projeto foram realizadas, tais como, a utilização do open closed principle para a realização do cálculo do imposto, visto que pode existir variações na forma de cálculo deste, para isso supus que o cálculo atual para livros é feito por uma constante vezes a diferença de preço de compra e venda mais outra constante vezes um valor relacionado ao gênero do livro. Outra decisão foi a utilização de classes abstratas, como Genero e Produto, para facilitar a implementação de outras classes. Além disso, as funções relacionadas com os produtos na classe livraria realizam operações no estoque, mas também chamam funções do produto, passando a livraria como argumento, isso foi feito para que fosse possível realizar todas as operações necessárias para concluir a tarefa, como por exemplo quando se adiciona um livro, é necessário não só mexer no estoque mas também na lista de livros de um autor. Ademais, a função alterar produto foi feita de forma a ser possível alterar qualquer um de seus parâmetros, por meio da passagem de key args. Por fim, foi criado um contador no cliente, utilizado para a criação e identificação das ordens de compra.

Após a realização do diagrama, o código correspondente foi escrito, onde alguns testes foram feitos, os quais, tanto o código como a saída obtida, estão expostos nas figuras abaixo.

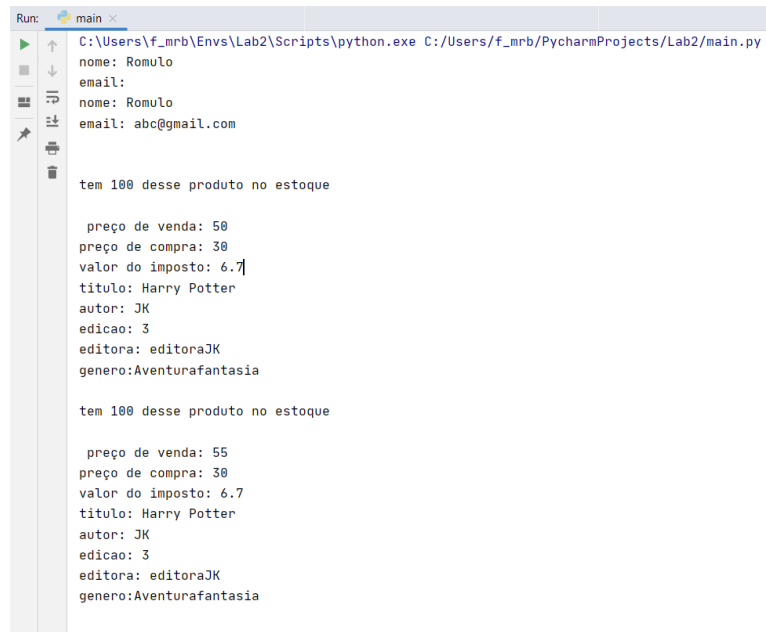
```
saraiva = Livraria("Saraiva")
calc = ImpostoLivroBR2022()
JK = Autor("JK")
HP = Livro("Harry Potter", 50, 30, 0, calc, JK, aventura_fantasia, 3, "editoraJK")
HP.setImposto()

saraiva.inserirCliente("Romulo")
saraiva.consultarCliente("Romulo")
saraiva.alterarCliente("Romulo", "abc@gmail.com")
Romulo = saraiva.consultarCliente("Romulo")

print("\n")

saraiva.inserirProduto(HP, 100)
saraiva.consultarProduto(HP)
saraiva.alterarProduto(HP, preco_venda=55)
saraiva.consultarProduto(HP)
```

Figura 2: Primeira metade do código de teste



```
Run: main x
C:\Users\f_mrb\Envs\Lab2\Scripts\python.exe C:/Users/f_mrb/PycharmProjects/Lab2/main.py
nome: Romulo
email:
nome: Romulo
email: abc@gmail.com

tem 100 desse produto no estoque

    preço de venda: 50
    preço de compra: 30
    valor do imposto: 6.7
    titulo: Harry Potter
    autor: JK
    edicao: 3
    editora: editoraJK
    genero:Aventurafantasia

tem 100 desse produto no estoque

    preço de venda: 55
    preço de compra: 30
    valor do imposto: 6.7
    titulo: Harry Potter
    autor: JK
    edicao: 3
    editora: editoraJK
    genero:Aventurafantasia
```

Figura 3: Primeira metade da saída do teste

```
saraiva.consultarAutor("JK")
```

```
saraiva.inserirOrdemCompra({HP:3}, Romulo)
saraiva.consultarOrdemCompra(1,Romulo)
saraiva.inserirOrdemCompra({HP:5}, Romulo)
saraiva.consultarOrdemCompra(2,Romulo)
saraiva.alterarOrdemCompra({HP:4}, 1, Romulo)
saraiva.consultarOrdemCompra(1,Romulo)
saraiva.removerOrdemCompra(2,Romulo)
saraiva.consultarOrdemCompra(2,Romulo)
```

```
print("\n")
```

```
saraiva.removerProduto(HP)
saraiva.consultarProduto(HP)
```

```
print("\n")
```

```
saraiva.removerCliente("Romulo")
saraiva.consultarCliente("Romulo")
```

Figura 4: Segunda metade do código de teste

```
Run: main x
genero:Aventurafantasia

o autor JK tem os seguintes livros:

preço de venda: 55
preço de compra: 30
valor do imposto: 6.7
titulo: Harry Potter
autor: JK
edicao: 3
editora: editoraJK
genero:Aventurafantasia

Compra 1
produto:Harry Potter, quantidade:3
Compra 2
produto:Harry Potter, quantidade:5
Compra 1
produto:Harry Potter, quantidade:4
ordem de compra 2 removida
não existe a ordem de compra 2

esse produto não esta no estoque

cliente Romulo removido
esse cliente não existe

Process finished with exit code 0
```

Figura 5: Segunda metade da saída do teste