

TRANSPORTADORA

O projeto tem como intuito realizar os procedimentos básicos requeridos em um sistema de gerência para uma transportadora, que atende tanto cliente pessoa física quanto cliente pessoa jurídica.

Foram utilizadas 5 classes:

- 'Cidade': Responsável por armazenar todas as informações em relação as capitais do Brasil, que serão utilizadas para os cálculos de valores e de tempo de entrega na classe 'Entrega'. Utiliza como base um sistema de coordenadas polares, com base em Belo Horizonte, que foi escolhida como origem. Os cálculos de distância são feitos com a transformação das coordenadas em cartesianas, resultando assim na distância aérea entre qualquer par de cidades.
- 'Entrega': A classe 'Entrega' é responsável por tudo em relação aos pedidos de entrega, como cadastro de novas entregas, consulta de preço e consulta do tempo de entrega. Ela não herda a classe 'Cidade', porém é altamente dependente dela, visto que possui como dois de seus parâmetros cidades. Dessa forma, as funções que calculam preços e distâncias utilizam das cidades recebidas para chamar funções da classe 'Cidade' e, assim, realizar os cálculos necessários com base na distância recebida.
- 'Cadastro': É a classe principal com relação aos cadastros do sistema. É a responsável por guardar e manipular dados essenciais, como nome, cpf e email, de todos os tipos de cadastro. Cadastra apenas pessoas físicas.
- 'CadastroPJ': É responsável pelos cadastros de pessoas jurídicas. Herda a classe 'Cadastro' e utiliza de todos os seus parâmetros, com exceção da função responsável por imprimir os dados do cliente, que recebe um overload, já que as informações recebidas terão outro tipo de identificação. (Ex.: cpf -> cnpj)
- 'CadastroFuncionario': Classe responsável por criar e manipular os cadastros dos funcionários da empresa. Também herda a classe 'Cadastro' e utiliza-se de todas suas funções e variáveis. Porém, alguns dados novos são adicionados, portanto são criadas novas funções get e set específicas para essa classe. Dessa forma, também é criada uma nova função para imprimir os parâmetros, que irá chamar a função de imprimir da classe 'Cadastro' e adicionará novos componentes.

Obs.: Com relação ao encapsulamento, todas os parâmetros das classes foram implementados como privados, para que haja maior proteção em relação a eles já que nenhuma outra classe poderá acessa-los. O conceito de herança foi utilizado em diversas classes, reduzindo a implementação de funções desnecessárias, assim simplificando o código e seu entendimento. Já o polimorfismo foi pouco necessário e utilizado apenas uma vez, a partir de uma função virtual, porém se fez útil para facilitar a implementação do código na função main, já que a mesma função representa ações diferentes dependendo de qual classe a chama.

Com relação a execução do programa, o container 'map' foi utilizado com frequência, tanto para armazenar os diferentes tipos de cadastro quanto para armazenar as cidades e as entregas. Isso foi feito para aproveitar a agilidade do seu sistema de busca. O tratamento de exceções foi utilizado diversas vezes durante todo o projeto, para garantir que as informações digitadas pelo usuário não causassem erros durante a execução do programa.