



Universidade Federal de Santa Catarina
Campus Araranguá - ARA
Centro de Ciências, Tecnologias e Saúde
Departamento de Computação
PLANO DE ENSINO

I. IDENTIFICAÇÃO DA DISCIPLINA

CÓDIGO	NOME DA DISCIPLINA	Nº DE HORAS-AULA SEMANAIS - TEÓRICAS	Nº DE HORAS-AULA SEMANAIS - PRÁTICAS
DEC7532	Linguagem de Programação II	0	4
TOTAL DE HORAS - AULA SEMESTRAIS	HORÁRIO TURMAS TEÓRICAS	HORÁRIO TURMAS PRÁTICAS	MODALIDADE
72	02655A - 2.1620-2	02655A - 4.1620-2	Presencial

ATIVIDADE PRÁTICA T1 – Trabalho Individual

Considere os seguintes *headers* (arquivos *.h* ou *.hpp*) das classes *Route* e *Vehicle* abaixo:

```
class Route
{
private:
    string *m_origin, *m_destination;
    float *m_distanceKm;
public:
    Route(string ori, string dest, float km);
    ~Route();

    string GetOrigin();
    string GetDestination();
    float GetDistanceKM();
};

class Vehicle
{
private:
    string m_plate, m_model, m_brand;
    float m_initialKm;
    vector<Route> m_trips;
public:
    Vehicle(string plate, string model, string brand);
    void IncludeTrip(string origin, string destination, float distance);
    bool RemoveTrip(size_t index);
    string SearchTripBySubstring(string wordToSearch);

    string GetAllTrips();
    float GetConsumption();
};
```

Estas classes tratam de um sistema gerencial de uma empresa de logística, onde a sua frota é gerenciada em um sistema de inclusão de viagens. A classe **Route** consiste dos seguintes atributos: cidade de origem, destino, e a distância em quilômetros. A classe possui seus atributos todos de forma dinâmica, e devem ser mantidos desta forma (considerando uma integração com classes legadas). A classe **Vehicle** trata-se de um veículo em particular, com informações de atributos como : placa, modelo e marca. Possui também a quilometragem inicial, que não necessariamente é zero (0) (considerando que veículos usados podem compor a frota). Um veículo pode realizar várias viagens (atributo **m_trips**), quantas forem necessárias, assim como possibilitar a exclusão (por índice do vetor). Ainda, os métodos **SearchTripBySubstring** deve ser implementado, e ao entrar com uma string de busca, deve-se procurar esta string tanto na cidade de origem, quando destino.

Pede-se:

Dar continuidade ao sistema segundo a sua interpretação, mantendo os headers conforme proposto (sem alterá-los), e implementando em arquivos de source (.cpp) a continuidade do programa. Nenhum dos atributos e métodos devem ser removidos, **mas podem ser inseridas novas informações e métodos adicionais**.

O objetivo é produzir um sistema funcional, onde possam ser cadastrados veículos, removidos, e os mesmos serem colocados a realizar viagens. Implementar o menu abaixo:

1. Cadastrar Veículo
2. Remover Veículo
3. Adicionar uma Rota a um Veículo
4. Remover uma Rota a um Veículo
5. Relatório por Veículo (pesquisar por placa, mostrar todas as rotas)
6. Relatório Geral (mostrar todos os veículos e quilometragem total)
7. Pesquisar por substring (entrar com uma substring e trazer as opções encontradas)
0. Sair