Relatório DAW

Este projeto é composto por 6 ficheiros principais:

* Program.cs ficheiro com a função main() ;
* Car.cs – contem a classe instância Car, que cria instancias do tipo Car com os atributos ‘id’, ‘parked’ e ‘time’;
* Time.cs – gera instancias do tipo Time relativas ao tempo atual quando o construtor por omissão é utilizado ou o tempo de saída do estacionamento pelo construtor por parâmetros;
* Zone.cs – contem a class zone que pode ser instanciada 3 vezes para definir as zonas do parquímetro. Tem ainda métodos para atribuir lugares aos carros, assim como libertar lugares e imprimir o ticket de estacionamento
* zoneTime – é onde o calculo do tempo de saída do estacionamento é feito e os horários de funcionamento definidos
* MyFunctions.cs – contem métodos estáticos acessórios para cálculos intermédios como calculo de faturação e incumprimento de prazos de estacionamento

Funcionalidades implementadas

Time:

* Foram implementados dois construtores para a classe instancia Time, um por omissão e outro por parâmetros. Isto foi feito para reutilizar a classe para as circunstancias em que é necessário o tempo atual assim como quando é necessário criar um objeto tempo de saída do estacionamento. Quando o construtor por omissão é utilizado, é criado um objeto do tipo tempo para representar o tempo atual, enquanto que o construtor por parâmetros recebe um array com a informação necessária para criar um objeto do tipo tempo que representa o instante temporal em que o estacionamento de um certo carro termina.

Zone:

* Um objeto do tipo Zona foi criado para as zonas 1 a 3 referidas no enunciado. Para além das características de preço e tempo máximo, foi ainda implementado um array do tipo Car para representar os espaços disponíveis em cada zona. Estes espaços são determinados de forma aleatória dentro de um intervalo realista, e diminui consoante carros são estacionados. Quando um carro é estacionado numa zona é disponibilizado um ticket com a hora de saída para esse carro e utilizando o id do ticket é possível remover o carro do estacionamento, o que liberta um lugar na zona.

ZoneTime:

* Dependendo dos valores associados com cada zona, foi implementada uma função para determinar o numero de minutos correspondentes ao dinheiro inserido pelo utilizador. Se a zona tiver uma duração máxima de estacionamento essa será o valor máximo de minutos que o utilizador pode comprar.
* Estes minutos são depois convertidos em horas e minutos e adicionados ao tempo corrente de forma a determinar o tempo de saída do estacionamento. Visto que o tempo não é pago fora do horário de funcionamento e caso sobrem minutos estes são usados no tempo de funcionamento seguinte, foi implementado uma função que toma isto em consideração para o calculo final do tempo. O raciocínio adotado foi que tempo para além do horário de funcionamento do dia em que o dinheiro foi inserido é considerado horas extra. Enquanto existem horas extra estas são distribuídas pelo dia seguinte. Se esse dia é de semana é verificado se existem mais de 11 horas extra para saber se a hora final será adiantada mais um dia, se for menor as horas extra são adicionadas às 9h00 para obter a hora final. O mesmo é feito para sábado, mas para 5 horas em vez de 11. Nos sábados, caso sobrem horas extra, o programa avança dois dias em vez de um de forma a ignorar o domingo, e o dia da semana faz restart para segunda-feira.
* Foi ainda implementado um array de dias existentes em cada mês para que o programa detete o último dia de um mês para saber que tem de avançar para o mês seguinte.
* Quando o parquímetro é utilizado no domingo, ele exibe apenas uma mensagem que informa o utilizador que não funciona nos domingos.

MyFunctions:

* Falar do menu
* Foi implementada uma função que é chamada quando o administrador acede ao seu menu e quer verificar se existem carros a exceder o limite de estacionamento. Esta itera todos os espaços de cada zona para verificar se existe algum carro estacionado que tem um tempo de saída inferior ao tempo atual, indicativo de que excedeu o seu tempo de funcionamento. Caso encontre expõe ao administrador o lugar do carro e a zona para que este possa tomar uma ação apropriada e de forma a identificar corretamente o carro.
* A faturação diária é feita de uma forma simples: sempre que uma transação é concluída o valor de dinheiro inserido é adicionado a uma variável. Por outro lado, sempre que após uma transação é devolvido troco, este valor é subtraído da mesma variável. Assim quando o administrador for verificar o dinheiro que foi ganho nesse dia vê o valor correto.

Funcionalidades extra

* Foi implementado um algoritmo de troco que é ativado nas zonas que têm limite de tempo. Se o utilizador colocar um valor superior ao necessário para comprar o tempo máximo da zona, a diferença é devolvida como troco.