1)

Implemente a classe Estacionamento com os seguintes atributos:

private string nome;

private int numVagasLivres;

private string[] vagas;

a) Implemente os seguintes métodos:

• Estacionamento(string nome, int numTotalVagas)

Método construtor que inicializa o atributo nome com o valor passado por parâmetro. O atributo **numVagasLivres** deverá ser inicializado com o número total de vagas do estacionamento. Instancie o vetor **vagas**;

• int Estacionar(string placa)

Se existirem vagas disponíveis no estacionamento, encontrar a primeira posição vazia do vetor vagas e inserir a placa informada como parâmetro. O método deve retornar o número da vaga (isto é a posição do vetor que a placa foi inserida). Se não existirem vagas disponíveis, esse método deve retornar -1;

• int BuscarNumVaga (string placa)

Retorna o número da vaga, ou seja, a posição do vetor vagas, ocupada pelo veículo com a placa informada como parâmetro. Se a placa informada não for encontrada, deve retornar -1;

• void Retirar(string placa)

Retira o veículo cuja placa corresponde à informada como parâmetro para esse método, marcando a vaga correspondente com null;

• Propriedade NumVagasLivres

Deve ter apenas o método get para retornar o número de vagas disponíveis no estacionamento;

• void ExibirOcupacao()

Imprime o número de cada vaga do estacionamento e a placa do veículo que a ocupa ou a informação de quea vaga está vazia.

b) Implemente a classe Teste, com o método Main e as seguintes manipulações:

• Crie um objeto Estacionamento com 20 vagas

• Estacione 4 carros com as seguintes placas: HKT0098, OLP4290, HJB0495 e OWB3904

• Exiba a ocupação do estacionamento

• Exiba o número da vaga onde o carro de placa HKT0098 está estacionado.

• Retire do estacionamento o veículo de placa HKT0098"

• Exiba a ocupação do estacionamento

• Estacione 3 carros com as seguintes placas: HTP5619, BOL4861 e HGT9436

• Exiba a ocupação do estacionamento

• Exiba a quantidade de vagas livres no estacionamento

2)

A) Implemente a classe CartaoValeTransporte com os seguintes atributos:

• número (int) – número identificador do cartão vale-transporte

• saldo (double) – saldo do cartão

• bloqueado (bool) – indica se o cartão vale-transporte está bloqueado ou não.

Implemente os seguintes métodos:

• CartaoValeTransporte(int numero, double saldo): Construtor que inicializa os atributos numero e saldo com

os valores passados através dos parâmetros numero e saldo, respectivamente; e o atributo bloqueado com

false;

• Recarregar(double credito): Se o cartão vale-transporte estiver não estiver bloqueado e o valor a ser

creditado, passado como parâmetro para esse método, for positivo: incrementa o saldo do cartão vale-

transporte com o valor de crédito informado;

• Propriedade Saldo: apenas com o método get

• PagarPassagem(double tarifa): Se o cartão vale-transporte estiver não estiver bloqueado e houver saldo

suficiente: decrementa, do saldo atual do cartão vale-transporte, o valor da tarifa informada;

• Bloquear(): bloqueia o cartão vale-transporte (atributo deve receber true).

B) Implemente uma classe chamada Teste que contenha o método Main e faça as seguintes operações:

• Crie um cartao1 com número 337540, e saldo igual a R$100,00

• Crie um cartao2 com número 553066, e saldo igual a R$50,00

• Pague uma passagem no valor de R$5,50 com o cartao2

• Imprima o saldo do cartao2

• Recarregue o cartao1 com R$25,00

• Imprima o saldo do cartao1

• Bloqueei o cartao2

• Recarregue o cartao2 com R$35,00

• Imprima o saldo do cartao2

3)

Crie as classes “Cliente”, “ContaCorrente” e “Agencia” conforme abaixo:

• A classe Cliente possui os atributos privados “nome” e “CPF”, ambos do tipo String, e um atributo conta do

tipo ContaCorrente.

• A classe ContaCorrente tem os atributos privados número e dígito, ambos inteiros, o atributo agência do tipo Agencia e o atributo saldo (double). Crie também um método Depositar que receba um parâmetro double com o valor do depósito e aumente o saldo da conta. Crie também um método "Sacar" que receba um parâmetro double com o valor do saque e diminua o saldo da conta. A conta não pode ficar negativa. Crie, finalmente, um método "ConsultarSaldo" que retorna uma String contendo o número da conta corrente com dígito, o númeroda agência com dígito e o saldo da conta corrente.

• A classe Agencia tem os atributos privados nome (String), número e dígito do tipo int.

• Para testar crie uma classe CaixaEletronico, que irá conter o método Main. No Main instancie um cliente com

os seguintes dados:

Nome: Ademar da Silva

- CPF: 123231518-12

- Conta Corrente: 1234 Dígito: 4

- Agência: 7890 Dígito: 5

- Saldo Inicial: R$150.00

Faça as seguintes operações:

- sacar 140.0

- consultar saldo

- sacar 200.0

- consultar saldo

- depositar 25.45

4)

Escreva um método recursivo que receba dois inteiros positivos a e n e calcule a^n

$$(a^n = \underbrace{a \cdot a \cdot a \cdots a}_{n \text{ vezes}})$$

5)

Escreva um método recursivo que receba como parâmetro dois números inteiros positivos m e n, e retorne a soma

dos números no intervalo [m,n].

Exemplo: m= 4 e n=7,

soma = 4 + 5 + 6 + 7 = 22

6)

Implemente um método recursivo para converter um número decimal para binário (isto é, converter um número do Sistema Decimal para o Sistema Binário). O método deve receber como parâmetro um número decimal positivo e deve imprimir na tela esse número convertido para binário (Obs: o método não deve ter retorno, deve ser void).

Uma maneira simples de resolver o problema é dividir o número decimal sucessivamente por 2 e pegar o resto da i-ésima divisão, da direita para esquerda.

Exemplo1 - Para o número 12 temos:

12/2 = 6, resto 0

6/2 = 3, resto 0,

3/2 = 1, resto 1

1/2 = 0, resto 1

Portanto, o número 12 em binário é 1100

7)

O máximo divisor comum (MDC) de dois números inteiros x e y pode ser calculado usando-se uma definição recursiva:

MDC(x, y) = MDC(x − y, y) se, x > y .

Além disso, sabe-se que:

MDC(x,y) = MDC(y,x)

MDC(x,x) = x

Exemplo:

MDC(10,6) = MDC(4,6) = MDC(6,4) = MDC(2,4) = MDC(4,2) = MDC(2,2) = 2

Implemente um método recursivo que receba como parâmetro dois números inteiros, calcule e retorne o MDC de x e y.

8)

Faça um método recursivo que receba um array de inteiros e um número inteiro n indicando o tamanho do array.

O método deve retornar a quantidade de números negativos no array (O método deve ter obrigatoriamente somente esses dois parâmetros).