OAT 1 CESAR

EXERCÍCIOS DE FIXAÇÃO 1

Questão 1: Qual a diferença entre Polimorfismo, Herança, Encapsulamento e Abstração?

Polimorfismo= É responsáveis pela especialização de classes em uma aplicação OO, pois em comportamento definidos, mas não implementados nas classes mais genéricas, serão implementadas nãos novas subclasses.

Herança= É uma nova classe, que pode ser criada a partir de outra existente, pode herdar atributos e comportamento da classe a ser estendida.

Encapsulamento= dar segurança aos objetos mantendo-os controlados em relação em relação , seu nível de acesso, oculta os membros de uma classe do acesso exterior , diminui os malefícios causados pela interferência externa sobre os dados , ´porque isola parte dos códigos.

Abstração= Tem a habilidade de modelar característica do mundo real, determina como as características irão atuar ao receber informação.

Questão 2: O que é Associação, Agregação, Composição e Generalização?

Associação= Representa a relação existente entre objetos descreve o vínculo entre duas classes e geralmente determina que as instâncias de uma classe estão de alguma forma ligadas às instância da outra.

Agregação= Representa uma relação todo-parte entre o agregado e suas partes, as informações precisam ser complementadas por um objetos de outra partes assim a perda do todo não resulta, obrigatoriamente na perda das partes.

Composição= tem um vínculo mais forte entre objetos -parte e objeto-todo elas tem um forte relacionamento , em que as partes não sobrevivem sem o todo.

Generalização= Representa a relação existente entre classes, identifica superclasse e sub-classe os métodos definidos no classe-mãe pelas classes filhas.

EXERCÍCIOS DE FIXAÇÃO 2

Questão 1: Crie uma classe de nome Carro e atribua a ela todas as propriedades que você acredita que um carro possua. Instancie a classe Carro e preencha 3 objetos distintos. Imprima na tela do usuário todos os atributos dos três carros.

class Carro

{

public string Marca { get; set;}

public string Modelo { get; set;}

public int Ano { get; set; }

public string Cor { get; set;}

public string Placa { get; set;}

}

class program

{

static void Main(string[] args)

{

Carro carro1 = new Carro();

carro1.Marca = "Fiat";

carro1.Modelo = "mobe";

carro1.Ano = 2019;

carro1.Cor = "vermelho";

carro1.Placa = "bra2e19";

Carro carro2 = new Carro();

carro2.Marca = "renault";

carro2.Modelo = "sandero";

carro2.Ano = 2020;

carro2.Cor = "azul";

carro2.Placa = "rio2a18";

Carro carro3 = new Carro();

carro3.Marca = "nissan";

carro3.Modelo = "sentra";

carro3.Ano = 2021;

carro3.Cor = "preto";

carro3.Placa = "bra0s17";

Console.WriteLine("CARRO1:");

Console.WriteLine("Marca:" + carro1.Marca);

Console.WriteLine("Modelo:" + carro1.Modelo);

Console.WriteLine("Ano:" + carro1.Ano);

Console.WriteLine("Cor:" + carro1.Cor);

Console.WriteLine("Placa:" + carro1.Placa);

Console.WriteLine();

Console.WriteLine("CARRO2:");

Console.WriteLine("Marca:" + carro2.Marca);

Console.WriteLine("Modelo:" + carro2.Modelo);

Console.WriteLine("Ano:" + carro2.Ano);

Console.WriteLine("Cor:" + carro2.Cor);

Console.WriteLine("Placa:" + carro2.Placa);

Console.WriteLine();

Console.WriteLine("CARRO3:");

Console.WriteLine("Marca:" + carro3.Marca);

Console.WriteLine("Modelo:" + carro3.Modelo);

Console.WriteLine("Ano:" + carro3.Ano);

Console.WriteLine("Cor:" + carro3.Cor);

Console.WriteLine("Placa:" + carro3.Placa);

Console.WriteLine();

}

}

Questão 2: Pesquise como um criar um método específico dentro de uma classe em C#.

Crie dois métodos para a classe Carro criada na questão anterior. Um método chamado

Acelerar(), que retorna a string “O carro está em movimento” como resposta e outro método

Freiar() que também retorna uma string “O carro está parado” como resposta.

class Carro

{

public string Marca { get; set; }

public string Modelo { get; set; }

public int Ano { get; set; }

public string Cor { get; set; }

public string Placa { get; set; }

public int VelocidadeMaxima { get; set; }

public void Acelerar()

{

Console.WriteLine("O carro está acelerando.");

}

public void Frear()

{

Console.WriteLine("O carro está freando.");

}

}

class Program

{

static void Main()

{

Carro carro1 = new Carro()

{

Marca = "Volkswagen",

Modelo = "Gol Quadrado",

Ano = 1980,

Cor = "Preto",

Placa = "bra2e19",

VelocidadeMaxima = 170

};

Carro carro2 = new Carro()

{

Marca = "Volkswagen",

Modelo = "Brasília",

Ano = 1973,

Cor = "Amarela",

Placa = "rio2a18",

VelocidadeMaxima = 140

};

Carro carro3 = new Carro()

{

Marca = "Volkswagen",

Modelo = "Variant",

Ano = 1969,

Cor = "Amarela",

Placa = "bra0s17",

VelocidadeMaxima = 1120

};

Console.WriteLine("Carro 1:");

Console.WriteLine("Marca: " + carro1.Marca);

Console.WriteLine("Modelo: " + carro1.Modelo);

Console.WriteLine("Ano: " + carro1.Ano);

Console.WriteLine("Cor: " + carro1.Cor);

Console.WriteLine("Placa:" + carro1.Placa);

Console.WriteLine("Velocidade Máxima: " + carro1.VelocidadeMaxima);

carro1.Acelerar();

carro1.Frear();

Console.WriteLine("Carro 2:");

Console.WriteLine("Marca: " + carro2.Marca);

Console.WriteLine("Modelo: " + carro2.Modelo);

Console.WriteLine("Ano: " + carro2.Ano);

Console.WriteLine("Cor: " + carro2.Cor);

Console.WriteLine("Placa:" + carro2.Placa);

Console.WriteLine("Velocidade Máxima: " + carro2.VelocidadeMaxima);

carro2.Acelerar();

carro2.Frear();

Console.WriteLine("Carro 3:");

Console.WriteLine("Marca: " + carro3.Marca);

Console.WriteLine("Modelo: " + carro3.Modelo);

Console.WriteLine("Ano: " + carro3.Ano);

Console.WriteLine("Cor: " + carro3.Cor);

Console.WriteLine("Placa:" + carro3.Placa);

Console.WriteLine("Velocidade Máxima: " + carro3.VelocidadeMaxima);

carro3.Acelerar();

carro3.Frear();

}

}

EXERCÍCIOS DE FIXAÇÃO 3

Questão 1: Quais são os benefícios de se criar um diagrama de classes?

• Ilustrar modelos de dados para sistemas de informação, não importa quão simples

ou complexo seja.

• Entender melhor a visão geral dos esquemas de uma aplicação.

• Expressar visualmente as necessidades específicas de um sistema e divulgar essas

informações por toda a empresa.

• Criar gráficos detalhados que destacam qualquer código específico necessário para

ser programado e implementado na estrutura descrita.

• Fornecer uma descrição independente de implementação de tipos utilizados em um

sistema e passados posteriormente entre seus componentes.

Questão 2: O que é um array e como é a sua implementação?

Array = É um conjunto de elementos de um mesmo tipo de dados onde cada elemento do conjunto é acessado pela posição no array que é dada através de um índice (uma sequência de números inteiros).

EXERCÍCIOS DE FIXAÇÃO 4

Questão 1: Faça um programa em C# (com a estrutura do...while) que leia 20 valores

inteiros e:

- Encontre e mostre o maior valor;

- Encontre e mostre o menor valor;

- Calcule e mostre a média dos números lidos;

int[] valores = new int[20];

int contador = 0;

int soma = 0;

while (contador < 20)

{

Console.WriteLine($" informe o {contador + 1}° valor:");

valores[contador] =

int.Parse(Console.ReadLine());

{

soma += valores[contador];

contador++;

}

}

Console.WriteLine();

int MenorValor = valores.Min();

Console.WriteLine($" o menor valor digita é:{MenorValor}");

int MaiorValor = valores.Max();

Console.WriteLine($" o maior valor digita é:{MaiorValor}");

double Media = (double)soma / 20;

Console.WriteLine($" a media dos valores digitados é:{Media}");

Console.ReadKey();

Questão 2: Faça o seguinte programa em C#. Uma loja utiliza o código V para compras à

vista e o código P para compras a prazo. Faça um algoritmo que recebe ao código (V ou P)

e o valor de 15 transações. Calcule e mostre:

- O valor total das compras à vista.

- O valor total das compras a prazo.

- O valor total das compras efetuadas.

int C = 0;

double totalV = 0, totalP = 0, totalC = 0;

while (C < 15)

{

Console.Write("informe o codigo da transacão (V para compras á vista ou P para compras á prazo):");

string COD =

Console.ReadLine().ToUpper();

Console.Write("informe o valor da transação:");

double valor =

double.Parse(Console.ReadLine());

if (COD == "V")

{

totalV += valor;

}

else if (COD == "P")

{

totalP += valor;

}

totalC += valor;

C++;

}

Console.WriteLine(" Total de compras á vista: R${0}", totalV);

Console.WriteLine(" Total de compra á prazo : R${0}", totalP);

Console.WriteLine(" Total de compras efetuadas: R${0}", totalC);

Console.ReadKey();

Questão 3: Faça o seguinte programa em C#. A prefeitura de Luziânia fez uma pesquisa

com 200 pessoas, coletando dados sobre o salário e o número de filhos. A prefeitura deseja

saber:

- A média do salário dessas pessoas.

- A média do número de filhos.

- O maior salário.

- O menor salário.

int totalPessoas = 200;

double somaSalario = 0;

int somaFilhos = 0;

double maiorSalario = 0;

double menorSalario = 0;

int countSalarioAte1500 = 0;

for (int i = 1; i <= totalPessoas; i++)

{

Console.Write("Digite o salário da pessoa #{0}: R$", i);

double salario = double.Parse(Console.ReadLine());

Console.Write("Digite o número de filhos da pessoa #{0}: ", i);

int numFilhos = int.Parse(Console.ReadLine());

if (i == 1)

menorSalario = salario;

somaSalario += salario;

somaFilhos += numFilhos;

if (salario > maiorSalario)

maiorSalario = salario;

if (salario < menorSalario)

menorSalario = salario;

if (salario <= 1500)

countSalarioAte1500++;

}

double mediaSalario = somaSalario / totalPessoas;

double mediaFilhos = (double)somaFilhos / totalPessoas;

double porcentagemSalarioAte1500 = (double)countSalarioAte1500 / totalPessoas \* 100;

Console.WriteLine("Média do salário: R$" + mediaSalario.ToString("F2"));

Console.WriteLine("Média do número de filhos: " + mediaFilhos.ToString("F2"));

Console.WriteLine("Maior salário: R$" + maiorSalario.ToString("F2"));

Console.WriteLine("Menor salário: R$" + menorSalario.ToString("F2"));

Console.WriteLine("Porcentagem de pessoas com salários até R$1500,00: " +

porcentagemSalarioAte1500.ToString("F2") + "%");

EXERCÍCIOS DE FIXAÇÃO 5

Questão 1: Faça um programa em C# que leia uma quantidade indefinida de objetos Carro,

composto pelos atributos, marca, valor, cor, modelo e ano, e:

- Ordene os carros pelo de maior valor;

- Imprima na tela todos os carros ordenados do maior valor para o de menor valor;

class Carro

{

public string Marca { get; set; }

public decimal Valor { get; set; }

public string Cor { get; set; }

public string Modelo { get; set; }

public int Ano { get; set; }

}

class Program

{

static void Main()

{

List<Carro> carros = new List<Carro>();

while (true)

{

Carro carro = new Carro();

Console.Write("Digite a marca do carro (ou digite 'sair' para encerrar): ");

string marca = Console.ReadLine();

if (marca.ToLower() == "sair")

break;

carro.Marca = marca;

Console.Write("Digite o valor do carro: ");

carro.Valor = decimal.Parse(Console.ReadLine());

Console.Write("Digite a cor do carro: ");

carro.Cor = Console.ReadLine();

Console.Write("Digite o modelo do carro: ");

carro.Modelo = Console.ReadLine();

Console.Write("Digite o ano do carro: ");

carro.Ano = int.Parse(Console.ReadLine());

carros.Add(carro);

Console.WriteLine();

}

carros.Sort((c1, c2) => c2.Valor.CompareTo(c1.Valor));

Console.WriteLine("Carros ordenados do maior valor para o menor valor:");

foreach (var carro in carros)

{

Console.WriteLine("Marca: " + carro.Marca);

Console.WriteLine("Valor: " + carro.Valor);

Console.WriteLine("Cor: " + carro.Cor);

Console.WriteLine("Modelo: " + carro.Modelo);

Console.WriteLine("Ano: " + carro.Ano);

}

}

}

Questão 2: Baseado no programa anterior (Questão 1) Crie uma interface para Cadastrar,

Excluir e Listar os carros.

- Cadastre um carro.

- Exclua um carro.

- Liste todos os carros do menor valor para o maior valor.

interface ICarroService

{

void CadastrarCarro(List<Carro> carros);

void ExcluirCarro(List<Carro> carros);

void ListarCarros(List<Carro> carros);

}

class Carro : ICarroService

{

public string Marca { get; set; }

public decimal Valor { get; set; }

public string Cor { get; set; }

public string Modelo { get; set; }

public int Ano { get; set; }

public void CadastrarCarro(List<Carro> carros)

{

Console.Clear();

Carro novoCarro = new Carro();

Console.Write("Digite a marca do carro: ");

novoCarro.Marca = Console.ReadLine();

Console.Write("Digite o valor do carro: ");

novoCarro.Valor = decimal.Parse(Console.ReadLine());

Console.Write("Digite a cor do carro: ");

novoCarro.Cor = Console.ReadLine();

Console.Write("Digite o modelo do carro: ");

novoCarro.Modelo = Console.ReadLine();

Console.Write("Digite o ano do carro: ");

novoCarro.Ano = int.Parse(Console.ReadLine());

carros.Add(novoCarro);

Console.Clear();

Console.WriteLine("Carro cadastrado com sucesso!");

Console.WriteLine("\_\_\_\_\_\_\_\_\_\_");

}

public void ExcluirCarro(List<Carro> carros)

{

Console.Write("Digite o modelo do carro que deseja excluir!");

string modeloCarro = Console.ReadLine();

bool carroEncontrado = false;

for (int i = 0; i < carros.Count; i++)

{

if (carros[i].Modelo.Equals(modeloCarro))

{

carros.RemoveAt(i);

carroEncontrado = true;

break;

}

}

if (carroEncontrado)

{

Console.Clear();

Console.WriteLine("Carro excluído com sucesso!");

}

else

{

Console.Clear();

Console.WriteLine("Modelo do carro inválido!");

}

}

public void ListarCarros(List<Carro> carros)

{

Console.Clear();

Console.WriteLine("Lista de carros cadastrados:");

for (int i = 0; i < carros.Count; i++)

{

Console.WriteLine($"Carro #{i + 1}");

Console.WriteLine("Marca: " + carros[i].Marca);

Console.WriteLine("Valor: " + carros[i].Valor);

Console.WriteLine("Cor: " + carros[i].Cor);

Console.WriteLine("Modelo: " + carros[i].Modelo);

Console.WriteLine("Ano: " + carros[i].Ano);

Console.WriteLine();

}

}

}

class Program

{

static void Main()

{

List<Carro> carros = new List<Carro>();

Carro carro = new Carro();

while (true)

{

Console.WriteLine("1 - Cadastrar carro");

Console.WriteLine("2 - Excluir carro");

Console.WriteLine("3 - Listar carros");

Console.WriteLine("0 - Sair");

Console.Write("Escolha uma opção: ");

int opcao = int.Parse(Console.ReadLine());

switch (opcao)

{

case 1:

carro.CadastrarCarro(carros);

break;

case 2:

carro.ExcluirCarro(carros);

break;

case 3:

carro.ListarCarros(carros);

break;

case 0:

Console.WriteLine("Encerrando o programa...");

return;

default:

Console.WriteLine("Opção inválida! Tente novamente.");

break;

}

}

}

}

Questão 3: Escreva um programa em C# que deverá ter as seguintes opções:

- Carregar Vetor.

- Listar Vetor.

- Exibir apenas os números pares do vetor.

- Exibir apenas os números ímpares do vetor.

- Exibir a quantidade de números pares existentes nas posições ímpares do vetor.

- Exibir a quantidade de números ímpares existentes nas posições pares do vetor.

- Sair

Observação: Deverá ser implementado um método para realizar cada uma das opções

acima.

class Program

{

static int[] vetor;

static int tamanho;

static void Main()

{

Console.Clear();

Console.WriteLine("Programa do Vetor");

while (true)

{

Console.WriteLine("1 - Carregar Vetor");

Console.WriteLine("2 - Listar Vetor");

Console.WriteLine("3 - Exibir números pares do vetor");

Console.WriteLine("4 - Exibir números ímpares do vetor");

Console.WriteLine("5 - Quantidade de pares em posições ímpares");

Console.WriteLine("6 - Quantidade de ímpares em posições pares");

Console.WriteLine("0 - Sair");

Console.Write("Escolha uma opção: ");

int opcao = int.Parse(Console.ReadLine());

switch (opcao)

{

case 1:

CarregarVetor();

Console.WriteLine("Vetor carregado com sucesso!");

break;

case 2:

ListarVetor();

break;

case 3:

ExibirPares();

break;

case 4:

ExibirImpares();

break;

case 5:

QuantidadeParesPosicoesImpares();

break;

case 6:

QuantidadeImparesPosicoesPares();

break;

case 0:

Console.WriteLine("Encerrando o programa...");

return;

default:

Console.WriteLine("Opção inválida! Tente novamente.");

break;

}

}

}

static void CarregarVetor()

{

Console.Write("Digite o tamanho do vetor: ");

tamanho = int.Parse(Console.ReadLine());

vetor = new int[tamanho];

for (int i = 0; i < tamanho; i++)

{

Console.Write("Digite o valor para a posição {0}: ", i);

vetor[i] = int.Parse(Console.ReadLine());

}

}

static void ListarVetor()

{

Console.Clear();

Console.WriteLine("Vetor:");

for (int i = 0; i < tamanho; i++)

{

Console.WriteLine("Posição {0}: {1}", i, vetor[i]);

}

}

static void ExibirPares()

{

Console.Clear();

Console.WriteLine("Números pares do vetor:");

for (int i = 0; i < tamanho; i++)

{

if (vetor[i] % 2 == 0)

{

Console.WriteLine(vetor[i]);

}

}

}

static void ExibirImpares()

{

Console.Clear();

Console.WriteLine("Números ímpares do vetor:");

for (int i = 0; i < tamanho; i++)

{

if (vetor[i] % 2 != 0)

{

Console.WriteLine(vetor[i]);

}

}

}

static void QuantidadeParesPosicoesImpares()

{

int count = 0;

for (int i = 1; i < tamanho; i += 2)

{

if (vetor[i] % 2 == 0)

{

count++;

}

}

Console.Clear();

Console.WriteLine("Quantidade de números pares em posições ímpares: {0}", count);

}

static void QuantidadeImparesPosicoesPares()

{

int count = 0;

for (int i = 0; i < tamanho; i += 2)

{

if (vetor[i] % 2 != 0)

{

count++;

}

}

Console.Clear();

Console.WriteLine("Quantidade de números ímpares em posições pares: {0}", count);

}

}