**Mariana Morais dos Santos**

**Exercício 1**

**Programa**

using System;

using System.Globalization;

namespace exercicio1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Digite quantos retângulos deseja: ");

int quantidade = int.Parse(Console.ReadLine());//Pergunta e armazena a quantidade de retângulos que deseja

Retangulo[] retangulos = new Retangulo[quantidade];//Declara e instancia o vetor

for (int i = 0; i < quantidade; i++)//Inicia uma repetição conforme a quantidade de retângulos que colocou

{

Console.WriteLine("Informe a altura do retângulo: ");

double Altura = double.Parse(Console.ReadLine());//Pergunta e armazena a altura do retângulo

Console.WriteLine("Informe a largura do retângulo: ");

double Largura = double.Parse(Console.ReadLine());//Pergunta e armazena a largura do retângulo

retangulos[i] = new Retangulo(Altura, Largura);//Puxa os dados do override para exibir na tela

Console.WriteLine(retangulos[i]);

}

}

}

}

**Classe**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace exercicio1

{

class Retangulo

{

//Declaração das variaveis onde são guardados as informações.

public double Altura { get; set; }

public double Largura { get; set; }

public Retangulo(double altura, double largura)

{

Altura = altura;

Largura = largura;

}

public double CalcularArea()

{

return Altura \* Largura; //Uma função que faz a conta para caucular a area.

}

public double CalcularPer()//Uma função que faz a conta do perimetro.

{

return 2 \* (Largura + Altura);

}

public double CalcularDiag()//Uma função que faz a conta da diagonal.

{

return Math.Sqrt((Largura \* Largura) + (Altura \* Altura));

}

public override string ToString()//override serve para colocar uma caixa de mensagem que deseja, para depois puxar de um jeito mais simples para o progrma principal.

{

return "Altura...: " + Altura.ToString("F2")

+ "\n"

+ "Largura...: " + Largura.ToString("F2")

+ "\n"

+ "Área......: " + CalcularArea().ToString("F2")

+ "\n"

+ "Perímetro.: " + CalcularPer().ToString("F2")

+ "\n"

+ "Diagonal..: " + CalcularDiag().ToString("F2");

}

}

}

**Diagrama**

**retangulo.cs**

- Altura: double;

- Largura: double;

CalcularArea (): double;

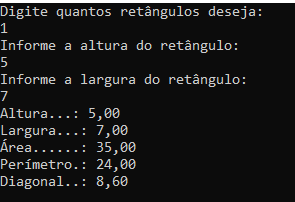
return Altura \* Largura;

CalcularPer (): double;

return 2 \* (Largura + Altura);

CalcularDiag (): double;

return Math.Sqrt((Largura \* Largura) + (Altura \* Altura));



**Exercício 2**

**Programa**

using System;

using System.Globalization;

namespace Exercicio2

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Digite quantos funcionários: ");

int quantidade = int.Parse(Console.ReadLine());// Pergunta e armazena a quantidade de funcionários que deseja

Dados\_Funcionario[] funcionario = new Dados\_Funcionario[quantidade];//Declara e instancia o vetor

for (int i = 0; i < quantidade; i++)//Inicia uma repetição conforme a quantidade de funcionários que colocou

{

Console.WriteLine("INFORME SEUS DADOS!!! ");

Console.WriteLine();

Console.WriteLine("NOME: ");

string nome = Console.ReadLine();// Pergunta e armazena o nome do funcionários

Console.WriteLine();

Console.WriteLine("SALÁRIO: ");

double salarioBruto = double.Parse(Console.ReadLine());// Pergunta e armazena o salário bruto do funcionários

Console.WriteLine();

Console.WriteLine("IMPOSTO: ");

double imposto = double.Parse(Console.ReadLine());// Pergunta e armazena o imposto do funcionário

Console.WriteLine();

funcionario[i] = new Dados\_Funcionario(nome, salarioBruto, imposto);// Joga as informações para class

Console.WriteLine(funcionario[i]); //Puxa os dados do override para exibir na tela

Console.WriteLine();

Console.WriteLine("DIGITE A PORCENTAGEM DE AUMENTO DE SALÁRIO: ");

double porcentagem = double.Parse(Console.ReadLine());// Pergunta e armazena a porcentagem de aumento do salário do funcionário

funcionario[i].AumentarSalario(porcentagem);

Console.WriteLine();

Console.WriteLine("DADOS ATUALIZADOS!");

Console.WriteLine(funcionario[i]);//Puxa os dados do override para exibir na tela dos dados atualizados

}

}

}

}

**Classe**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;namespace Exercicio2

{

class Dados\_Funcionario

{

//Declaração das variaveis onde são guardados as informações.

public string Nome { get; set; }

public double SalarioBruto { get; set; }

public double Imposto { get; set; }

public Dados\_Funcionario(string nome, double salarioBruto, double imposto)

{

Nome = nome;

SalarioBruto = salarioBruto;

Imposto = imposto;

}

public double CalcularSalarioLiquido() //Uma função que faz a conta SalarioBruto - Imposto.

{

return SalarioBruto - Imposto;

}

public void AumentarSalario(double perct\_aumento)//Uma função que faz a conta para aumentar o selário

{

SalarioBruto = SalarioBruto + (SalarioBruto \* perct\_aumento / 100.00);

}

public override string ToString() //override serve para colocar uma caixa de mensagem que deseja, para depois puxar de um jeito mais simples para o progrma principal.

{

return "O funcionário " + Nome.ToUpper()

+ "\nRecebe R$ " + SalarioBruto.ToString("F2")

+ " de salário bruto. "

+ "\nCom

imposto R$ " + Imposto.ToString("F2")

+ "\nTendo salário liquído de R$ " + CalcularSalarioLiquido();

}

}

}

**Diagrama**

**funcionario.cs**

- Nome: string;

- SalarioBruto: double;

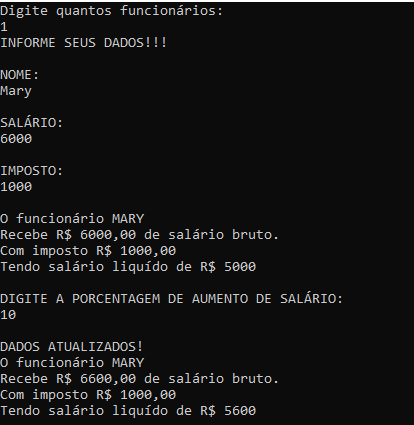
- Imposto: double;

CalcularSalarioLiquido(): double;

return SalarioBruto - Imposto;

AumentarSalario(): double;

SalarioBruto = SalarioBruto + (SalarioBruto \* perct\_aumento / 100.00);



**Exercício 3**

**Programa**

using System;

using System.Globalization;

namespace AV\_Alunos;

class Principal

{

static void Main(string[] args)

{

Console.Write("Digite a Quantidade de Alunos: ");

int quantidade\_alunos = int.Parse(Console.ReadLine());// Pergunta e armazena a quantidade de alunos que deseja.

Aluno[] alunos = new Aluno[quantidade\_alunos];//Declara e instancia o vetor

for (int i = 0; i < quantidade\_alunos; i++)//Inicia uma repetição conforme a quantidade de alunos que colocou.

{

Console.WriteLine();

Console.WriteLine("\*\*\* DADOS ALUNO " + (i + 1) + " \*\*\*");

Console.Write("Insira o Nome do Aluno: ");

string Nome = Console.ReadLine();// Pergunta e armazena o nome do aluno.

Console.WriteLine("Digite a nota do 1º trimestre : (limite maximo da nota é 30)");

double nota1 = double.Parse(Console.ReadLine());// Pergunta e armazena a nota do aluno.

while (nota1 > 30)//Começa repetição enquanto a nota for maior que 30.

{

if (nota1 > 30)//Se for maior que 30 mostrar a mensagem abaixo.

{

Console.WriteLine("Nota inválida, nota máxima é 30, digite novamente! ");

Console.WriteLine("Digite a nota do 1º trimestre : (limite maximo da nota é 30)");

nota1 = double.Parse(Console.ReadLine());

}

}

Console.WriteLine("Digite a nota do 2º trimestre : (limite maximo da nota é 35)");

double nota2 = double.Parse(Console.ReadLine());// Pergunta e armazena a segunda nota do aluno.

while (nota2 > 35)//Começa repetição enquanto a nota for maior que 35.

{

if (nota2 > 35)//Se for maior que 35 mostrar a mensagem abaixo.

{

Console.WriteLine("Nota inválida, nota máxima é 35, digite novamente! ");

Console.WriteLine("Digite a nota do 2º trimestre : (limite maximo da nota é 35)");

nota2 = double.Parse(Console.ReadLine());

}

}

Console.WriteLine("Digite a nota do 3º trimestre : (limite maximo da nota é 35)");

double nota3 = double.Parse(Console.ReadLine());// Pergunta e armazena a terceira nota do aluno.

while (nota3 > 35)//Começa repetição enquanto a nota for maior que 35.

{

if (nota3 > 35)//Se for maior que 35 mostrar a mensagem abaixo.

{

Console.WriteLine("Nota inválida, nota máxima é 35, digite novamente! ");

Console.WriteLine("Digite a nota do 3º trimestre : (limite maximo da nota é 35)");

nota3 = double.Parse(Console.ReadLine());

}

}

Console.WriteLine();

Console.WriteLine();

alunos[i] = new Aluno(Nome, nota1, nota2, nota3);

}

for (int i = 0; i < quantidade\_alunos; i++)//Inicia uma repetição conforme a quantidade de alunos disponíves.

{

Console.WriteLine();

Console.WriteLine(" \*\*\*\*\*\*\*\*\*\*\*\*NOTA FINAL DO " + (i + 1) + "° ALUNO \*\*\*\*\*\*\*\*\*\*\*");

Console.WriteLine();

Console.WriteLine(alunos[i]);

Console.WriteLine();

}

}

}

**Classe**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AV\_Alunos

{

class Aluno

{

public string Nome { get; set; }

public double Nota1 { get; set; }

public double Nota2 { get; set; }

public double Nota3 { get; set; }

public Aluno(string nome, double nota1, double nota2, double nota3)

{

Nome = nome;

Nota1 = nota1;

Nota2 = nota2;

Nota3 = nota3;

}

public double SomaNotas()

{

return Nota1 + Nota2 + Nota3;

}

public string ReprovadoAprovado()

{

if (SomaNotas() >= 60)

{

return "Parabéns voce foi APROVADO, sua nota final foi " + SomaNotas();

}

else

{

return "Aluno REPROVADO, sua nota final foi " + SomaNotas() + " faltaram " + (60 - SomaNotas()) + " pontos para ser aprovado!";

}

}

public override string ToString()

{

return "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DADOS DO ALUNO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

+ "Aluno(a) "

+ Nome.ToUpper() + "\n"

+ "\n Nota 1 "

+ Nota1.ToString("F2")

+ "\n Nota 2 "

+ Nota2.ToString("F2")

+ "\n Nota 3 "

+ Nota3.ToString("F2") + "\n\n"

+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*APROVADO OU REPROVADO\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

+ ReprovadoAprovado()

+ "\n\n";

}

}

}

**Diagrama**

**aluno.cs**

- Nome: string;

- Nota1: double;

-Nota2: double;

-Nota3: double;

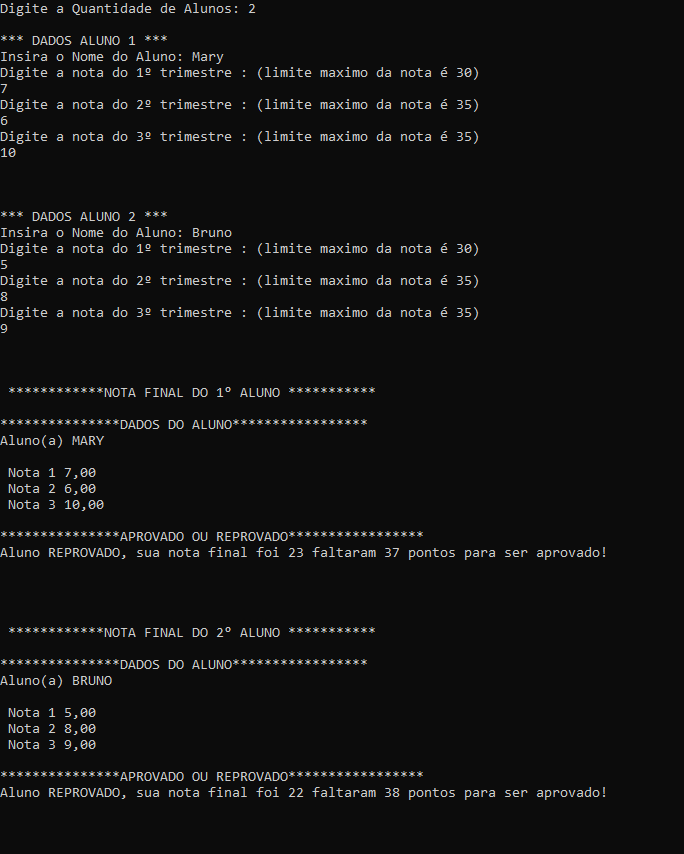
SomaNotas(): double;

return Nota1 + Nota2 + Nota3;

ReprovadoAprovado(): string;

return "Parabéns voce foi APROVADO, sua nota final foi " + SomaNotas();

return "Aluno REPROVADO, sua nota final foi " + SomaNotas() + " faltaram " + (60 - SomaNotas()) + " pontos para ser aprovado!";



**Exercício 4**

**Programa**

using System;

namespace quarto\_de\_aluguel

{

class Programa

{

static void Main(string[] args)

{

Console.WriteLine("SEJA BEM VINDO AO NOSSO SERVIÇO DE HOSPEDAGEM");

Console.WriteLine();

Console.WriteLine("Digite a quantidade de quartos disponíveis: ");

int quantidade\_quartos = int.Parse(Console.ReadLine()); //Pergunta e armazena a quantidade de quartos disponíveis

quarto quarto = new quarto();

dados[] quartos = new dados[quantidade\_quartos];//Declara e instancia o vetor

for (int i = 0; i < quantidade\_quartos; i++) //Inicia uma repetição conforme a quantidade de quartos disponíves

{

Console.WriteLine("\*\*\*\*\*\*DADOS DO HOSPEDE\*\*\*\*\*" + (i + 1));

Console.WriteLine(); Console.WriteLine("Digite seu nome: ");

string nome = Console.ReadLine();//Pergunta o nome do hóspede

Console.WriteLine("Digite seu enderço: ");

string endereco = Console.ReadLine();//Pergunta e armazena o endereço do hóspede

Console.WriteLine("Digite seu e-mail: ");

string email = Console.ReadLine();//Pergunta e armazena o Email do hóspede

Console.WriteLine();

Console.WriteLine("\*\*\*\*\*INFORMAÇÕES SOBRE O ALUGUEL\*\*\*\*\*\*");

Console.WriteLine("Digite o valor mensal do aluguel: ");

quarto.diaria = double.Parse(Console.ReadLine()); //Pergunta e armazena o valor mesal do aluguel

Console.WriteLine("Quantos meses deseja se hospedar em nosso quarto? ");

quarto.meses = int.Parse(Console.ReadLine()); //Pergunta e armazena a quantidade de mes que o hóspede vai se hóspedar

quartos[i] = new dados(nome, endereco, email); //Puxa os dados do override para exibir na tela

Console.WriteLine("O valor total do seu aluguel ficou: " + quarto.valor());

Console.WriteLine(quartos[i]);

}

Console.WriteLine();

Console.WriteLine("Pressione ENTER para encerrar.");

Console.ReadLine();

}

}

}

**Classe**

namespace Exercicio4

{

class dados

{

//Declaração das variaveis onde são guardados as informações.

public string Nome { get; set; }

public string Endereco { get; set; }

public string Email { get; set; }

public dados(string nome, string endereco, string email)

{

Nome = nome;

Endereco = endereco;

Email = email;

}

public override string ToString()//override serve para colocar uma caixa de mensagem que deseja, para depois puxar de um jeito mais simples para o progrma principal.

{

return Nome

+ ", "

+ Endereco

+ ", "

+ Email;

}

}

class quarto

{

public double diaria;

public int meses;

public double valor()

{

return diaria \* meses;

}

}

}

**Diagrama**

**Hospedagem.cs**

-Nome: string;

-Endereço: string;

-Email: string;

