https://github.com/MatheusComino/BP

ATIVIDADE 01

Classe: Retangulo

- Altura : double;

- Largura : double;

+ Retangulo(altura : double,largura double);

+ CalculaArea() : double;

+ CalculaPerimetro() : double;

+ CalculaDiagonal() : double;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DadosRetangulo

{

internal class Retangulo

{

public double altura, largura;

//METODO CONTRUTOR PARA PREENCHER AS VARIAVEIS

public Retangulo(double altura, double largura)

{

this.altura = altura;

this.largura = largura;

}

//SERVE PARA CALCULAR A AREA DO RETANGULO

public double calculaArea()

{

return altura \* largura;

}

//SERVE PARA CALCULAR O PERIMETRO DO RETANGULO

public double calculaPerimetro()

{

return (2 \* altura) + (2 \* largura);

}

//SERVE PARA CALCULAR A DIAGONAL DO RETANGULO

public double calculaDiagonal()

{

return Math.Sqrt((altura \* altura) + (largura \* largura));

}

}

}

using System;

using DadosRetangulo;

class program

{

static void Main()

{

//SERVE APRA PERGUNTAR A QUANTIDADE DE QUADRADOS

Console.Write("Quantidade de quadrados: ");

Retangulo[] retangulos = new Retangulo[int.Parse(Console.ReadLine())];

for (int i = 0; i < retangulos.Length; i++)

{

//SERVE PARA PEGUNTAR OS VALORES DE ALTURA E LARGURA

Console.Write("Altura do retangulo: ");

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

Console.Write("Largura do retangulo: ");

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

retangulos[i] = new Retangulo(altura, largura);

//SERVE PARA CALCULAR A AREA, PERIMETO E DIAGONAL

double area = retangulos[i].calculaArea();

double perimetro = retangulos[i].calculaPerimetro();

double diagonal = retangulos[i].calculaDiagonal();

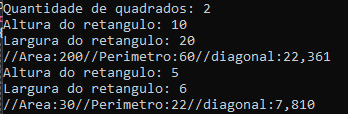
//SERVE PARA MOSTRAR OS VALORES DE AREA, PERIMETRO E DIAGONAL PARA O USUARIO.

Console.WriteLine("//Area:" + area + "//Perimetro:" + perimetro + "//diagonal:" + diagonal.ToString("F3"));

}

}

}



ATIVIDADE 02

Classe : Funcionário

- Nome : string;

- SalarioBruto : double;

- Imposto : double;

+ Funcionario(nome : string, salarioBruto : double, imposto : double)

+ CalculaSalarioLiquidos() : double;

+ CalculaAlmentosalario(percentual\_Almento : double) : double;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace FuncionarioSalario

{

internal class Funcionario

{

public string Nome;

public double SalarioBruto;

public double Imposto;

//METODO CONTRUTOR PARA PASSAR OS VALORES PARA A CLASSE

public Funcionario(string nome, double salarioBruto, double imposto)

{

Nome = nome;

SalarioBruto = salarioBruto;

Imposto = imposto;

}

//RETORNA O SALARIO SALARIO BRUNO MENOS O IMPOSTO

public double CalcularSalarioLiquidos()

{

return SalarioBruto - Imposto;

}

//RETORNA O SALARIO COM O REAJUSTE EM % JÁ DESCONTADO O IMPOSTO

public double CalcularAlmentoSalario(double percentual\_Almento)

{

return SalarioBruto += (SalarioBruto \* (percentual\_Almento / 100)) - Imposto;

}

//SERVE PARA TER UM TEXTO PADRÃO AO CHAMAR SOMENTE A CLASSE FUNCIONARIO

public override string ToString()

{

return "O funcionario " + Nome.ToUpper() + ", recebe R$" + CalcularSalarioLiquidos().ToString("F2") + " de pagamento mensal.";

}

}

}

using System;

using FuncionarioSalario;

using System.IO;

class Principal

{

static void Main(string[] args)

{

//SERVE APRA PERGUNTAR A QUANTIDADE DE FUNCIONARIOS

Console.Write("Quantidade de Funcionarios: ");

Funcionario[] funcionario = new Funcionario[int.Parse(Console.ReadLine())];

for (int i = 0; i < funcionario.Length; i++)

{

//PERGUNTA OS DADOS DO FUNCIONARIO

Console.WriteLine("\n[Entre com os dados do Funcionario]");

Console.Write("Nome:");

string nome = Console.ReadLine();

Console.Write("Salario Bruto:");

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

Console.Write("Imposto: ");

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

//PASSA OS VALORES PARA A CLASE FUNCIONARIO ATRAVEZ DE UM METODO CONTRUTOR

funcionario[i] = new Funcionario(nome, salarioBruto, imposto);

Console.WriteLine("Salario liquido: " + funcionario[i].CalcularSalarioLiquidos());

//PERGUNTA A QUANTIDADE DE REAJUSTE EM % DO SALARIO E MOSTRA O SALARIO REAJUSTADO

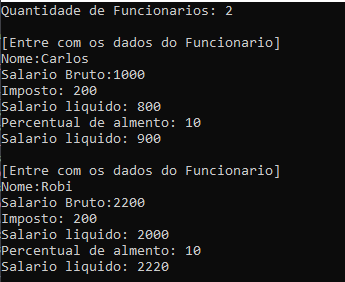
Console.Write("Percentual de almento: ");

Console.WriteLine("Salario liquido: " + funcionario[i].CalcularAlmentoSalario(double.Parse(Console.ReadLine())));

}

}

}



ATIVIDADE 03

Classe : Aluno

- nome : string;

- notas : double[3];

+ Aluno(nome : string, notas : double[])

+ CalculaNota() : double;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections;

namespace AlunoNota

{

internal class Aluno

{

public string nome;

public double[] notas = new double[3];

//METODO CONTRUTOR PARA PASSAR OS VALORES PARA AS VARIAVEIS

public Aluno(string nome, double[] notas)

{

this.nome = nome;

this.notas = notas;

}

//SERVE PARA SOMAR AS 3 NOTAS DO ALUNO

public double calculaNota()

{

double somaNotas = 0;

for (int i = 0; i < notas.Length; i++)

{

somaNotas += notas[i];

}

return somaNotas;

}

}

}

using System;

using AlunoNota;

class Program

{

static void Main()

{

//SERVE APRA PERGUNTAR A QUANTIDADE DE AlUNOS

Console.Write("Quantidade de alunos: ");

Aluno[] alunos = new Aluno[int.Parse(Console.ReadLine())];

for (int i = 0; i < alunos.Length; i++)

{

//SER PARA PERGUNTAR O NOME DO ALUNO

Console.Write("Nome do aluno: ");

string nome = Console.ReadLine();

Console.WriteLine("Digite as três notas do aluno: ");

//SERVE PARA GUARDAR AS NOTAS EM UM ARRAY

double[] notas = new double[3];

for (int j = 0; j < 3; j++)

{

Console.Write("Nota " + (j + 1) + ": ");

notas[j] = double.Parse(Console.ReadLine());

//SERVE PARA VERIFICAR SE A NOTAS DIGITADA É VALIDA

if (notas[j] > 30 || Double.IsNaN(notas[j]))

{

Console.WriteLine("Digite novamente, a nota precisa ser menor que 30");

j--;

}

else if (notas[j] >= 35 || notas[j] == null)

{

Console.WriteLine("Digite novamente, a nota precisa ser menor que 35");

j--;

}

}

//SERVE PARA PASSAR OS VALORES DIGITADOS PARA A CLASSE

alunos[i] = new Aluno(nome, notas);

//SERVE PARA MOSTRAR OS VALORES PARA O USUARIO

double somaNotas = alunos[i].calculaNota();

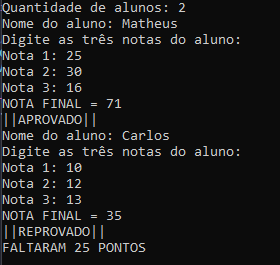
Console.WriteLine("NOTA FINAL = " + somaNotas);

Console.WriteLine(somaNotas >= 60 ? "||APROVADO||" : "||REPROVADO||\nFALTARAM " + (60 - somaNotas) + " PONTOS");

}

}

}



ATIVIDADE 04

- Classe : Quarto

- Numero : int;

- Tamanho : double;

- PrecoDiaria : double;

- QuantosDias : int;

- Locatario : Locatario;

+ Quarto(numero : int, tamanho : double, precoDiaria : double)

+ CalculaValorParaAlugar() : double;

- Classe: Locatario

Nome : string;

Cpf : string;

Idade : int;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

namespace QuartoAluguel

{

class Quarto

{

//Variaveis para guardar dados do quarto

public int numero;

public double tamanho;

public double precoDiaria;

public int quantosDias;

public bool estaAlugado;

public bool estaPago;

//serve para ter uma ligação do locatario com o quarto

public Locatario locatario = new Locatario();

//Serve para calcular o valor total para alugar o quarto.

public double CalculaValorParaAlugar()

{

return quantosDias \* precoDiaria;

}

//Metodo contrutor para criar novas intancias da classe;

public Quarto(int numero, double tamanho, double precoDiaria)

{

this.numero = numero;

this.tamanho = tamanho;

this.precoDiaria = precoDiaria;

}

//Seve para ter um texto padrão ao só chamar quato

public override string ToString()

{

return locatario

+ "\n[QUANTO]"

+ "\nNumero: " + numero

+ "\nTamanho: " + tamanho + "m²"

+ "\nPreço diaria: " + precoDiaria.ToString("C2")

+ "\nDias Alugados: " + quantosDias

+ "\nValor total: " + CalculaValorParaAlugar().ToString("C2");

}

}

class Locatario

{

//Serve para guardar os valores do locatario

public string nome;

public string cpf;

public int idade;

//Serve para mostrar um texto padrão ao só chamar locatario

public override string ToString()

{

return "[LOCATARIO]"

+ "\nNome:" + nome

+ "\nCpf:" + cpf

+ "\nIdade: " + idade;

}

}

}

using System;

using QuartoAluguel;

using System.IO;

class Program

{

static void Main()

{

//--Verifica se o existe um arquivo, se não cria um txt

string path = @"c:\temp\MyTest.txt";

if (!File.Exists(path))

{

using (StreamWriter sw = File.CreateText(path))

{

}

}

Quarto[] quarto = new Quarto[999];

Console.WriteLine("[1]. CADASTRAR UM NOVO QUARTO\n[2]. CONSULTAR TODOS OS QUARTOS\n[3]. ALUGAR UM QUARTO\n[4]. DADOS DOS LOCATARIOS\n[5]SAIR");

int switch\_on = int.Parse(Console.ReadLine());

//Serve para repetir o programa ate o usuario querer parar

while (switch\_on < 5)

{

//Cadastra um novo quarto

if (switch\_on == 1)

{

List<string> list = new List<string>();

//--Consulta os valores no documento

StreamReader sr = new StreamReader(path);

string line = sr.ReadLine();

while (line != null)

{

list.Add(line);

line = sr.ReadLine();

}

sr.Close();

//-- Adiciona dados no documento

StreamWriter sw = new StreamWriter(path);

Console.Write("NUMERO:");

list.Add(Console.ReadLine());

Console.Write("TAMANHO DO QUARTO:");

list.Add(Console.ReadLine());

Console.Write("PREÇO DA DIARIA:");

list.Add(Console.ReadLine());

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

{

sw.WriteLine(list[i]);

}

sw.Close();

}

//Mostra todos os quartos cadastrados

else if (switch\_on == 2)

{

StreamReader sr = new StreamReader(path);

string line = sr.ReadLine();

if (line == null) Console.WriteLine("NENHUM CARTO DISPONIVEL");

int i = 0;

while (line != null)

{

int nome = int.Parse(line);

double tamanho = double.Parse(sr.ReadLine());

double precoDiaria = double.Parse(sr.ReadLine());

line = sr.ReadLine();

quarto[i] = new Quarto(nome, tamanho, precoDiaria);

Console.WriteLine("[QUARTOS " + i + "]");

Console.WriteLine("Numero do Quarto: " + quarto[i].numero);

Console.WriteLine("Tamanho do Quarto: " + quarto[i].tamanho + "m²");

Console.WriteLine("Preço Diaria: " + quarto[i].precoDiaria.ToString("C2"));

i++;

}

sr.Close();

}

//Serve para o cliente alugar um quarto

else if (switch\_on == 3)

{

Console.Write("DIGITE O QUARTO:");

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

Console.Write("\nNome: ");

quarto[numeroQuarto].locatario.nome = Console.ReadLine();

Console.Write("Cpf: ");

quarto[numeroQuarto].locatario.cpf = Console.ReadLine();

Console.Write("Idade: ");

quarto[numeroQuarto].locatario.idade = int.Parse(Console.ReadLine());

Console.Write("Dias de locação: ");

quarto[numeroQuarto].quantosDias = int.Parse(Console.ReadLine());

Console.WriteLine("\nCustara: " + quarto[numeroQuarto].CalculaValorParaAlugar().ToString("C2"));

string confirmaLocacao;

Console.WriteLine("Confirmar? [S] ou [N]");

confirmaLocacao = Console.ReadLine();

if (confirmaLocacao == "S" || confirmaLocacao == "s")

{

quarto[numeroQuarto].estaAlugado = true;

Console.WriteLine("---------");

Console.WriteLine(quarto[numeroQuarto]);

}

else

{

Console.WriteLine("Obrigado, Até a proxima!!!");

}

}

//Serviria para ver os dados de todas as pessoas cadastradas porem não deu tempo de fazer

else if (switch\_on == 4)

{

Console.WriteLine("PROGRAMA EM PRODUÇÃO, FUNCIONALIDADE INDISPONIVEL");

}

Console.WriteLine("\n[1]. CADASTRAR UM NOVO QUARTO\n[2]. CONSULTAR TODOS OS QUARTOS\n[3]. ALUGAR UM QUARTO\n[4]. DADOS DOS LOCATARIOS\n[5]SAIR");

switch\_on = int.Parse(Console.ReadLine());

}

}

}

