

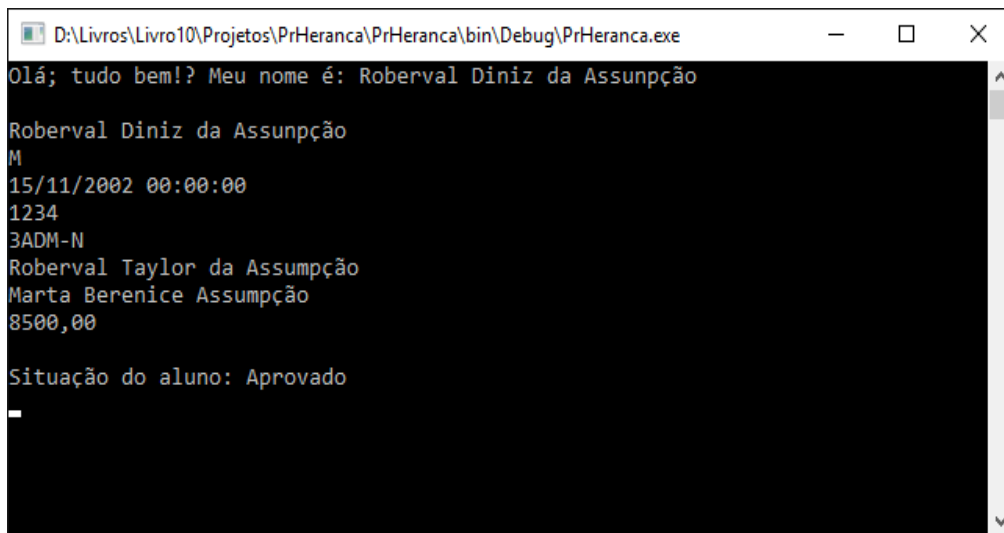
...

Herança é um conceito importantíssimo da OOP, pois, define um mecanismo muito poderoso de reutilização de código. Do ponto de vista prático, pode ser entendido como a ação de criar um subconjunto de instâncias a partir de um conjunto de instâncias, com os elementos desse subconjunto herdando todas as características do conjunto original. Na prática, o objetivo é fornecer um mecanismo simples - porém muito poderoso - para definir novas classes a partir de uma já existente. Deste modo, essas novas classes herdam todos os membros (*propriedades+métodos*) da classe original, tornando esse mecanismo uma técnica muito eficiente para construir, organizar e reutilizar código.

Na prática, a herança possibilita a criação de uma nova classe de modo que essa classe, denominada subclasse, classe-filha ou classe derivada, herda TODAS as características da classe-mãe (denominada superclasse, classe base ou classe ancestral) e podendo, ainda, adquirir *propriedades* e *métodos* próprios, sofrendo o que se chama de **especialização**. Assim sendo, fazendo uma comparação com a vida diária, este mecanismo permite criar “coisas” a partir de um modelo. De um modo geral, para aplicações em projetos tipo console em C#, a classe **Programa** é a classe principal que se apresenta para todos os novos projetos, e “dentro” dela o método **Main()** é a rotina que sempre é a primeira a ser executada quando a aplicação for executada (carregada) e pode gerar outras classes pelo mecanismo de herança. Por exemplo, criar a classe **TAluno**, derivada da classe **TPessoa**; a sintaxe em C# é a seguinte:

```
Class TAluno : TPessoa;
```

Então, a classe **TAluno** possuirá as mesmas propriedades e os mesmos métodos da classe **TPessoa** e, além disto, **TAluno** poderá adquirir outros membros além daqueles da sua classe primitiva. Isto quer dizer que a classe derivada poderá ter, por exemplo, outras propriedades como: *matrícula*, *turma*, *status*, etc, e também incorporar novos métodos para ações próprias, como, por exemplo, *Cadastrar* e *VerificarStatus* do aluno em função da média obtida. Veja a **figura 1**, que mostra o resultado de uma aplicação com código em C# console; e a **figura 2** mostra os membros (*propriedades+métodos*) da classe **TAluno** acrescidos dos membros herdados da classe **TPessoa**. Observe que não foi preciso reescrever o método *FunSaudar()* na classe **TAluno**, pois ele foi herdado da classe **TPessoa**. O mesmo aconteceu com as propriedades *Pessoa_nome*, *Pessoa_sexo* e *Pessoa_nasc*, que também foram herdadas da classe **TPessoa**. Isto é uma demonstração de que Herança é um conceito importantíssimo, pois economiza linhas de código reaproveitando os que já estão prontos. Neste caso, mostrou que um *aluno* é uma *pessoa*, e todas as pessoas (sejam elas alunos ou não) têm um *nome*, uma *data de nascimento* e um tipo de *sexo*, além de ter a capacidade de fazer uma saudação!

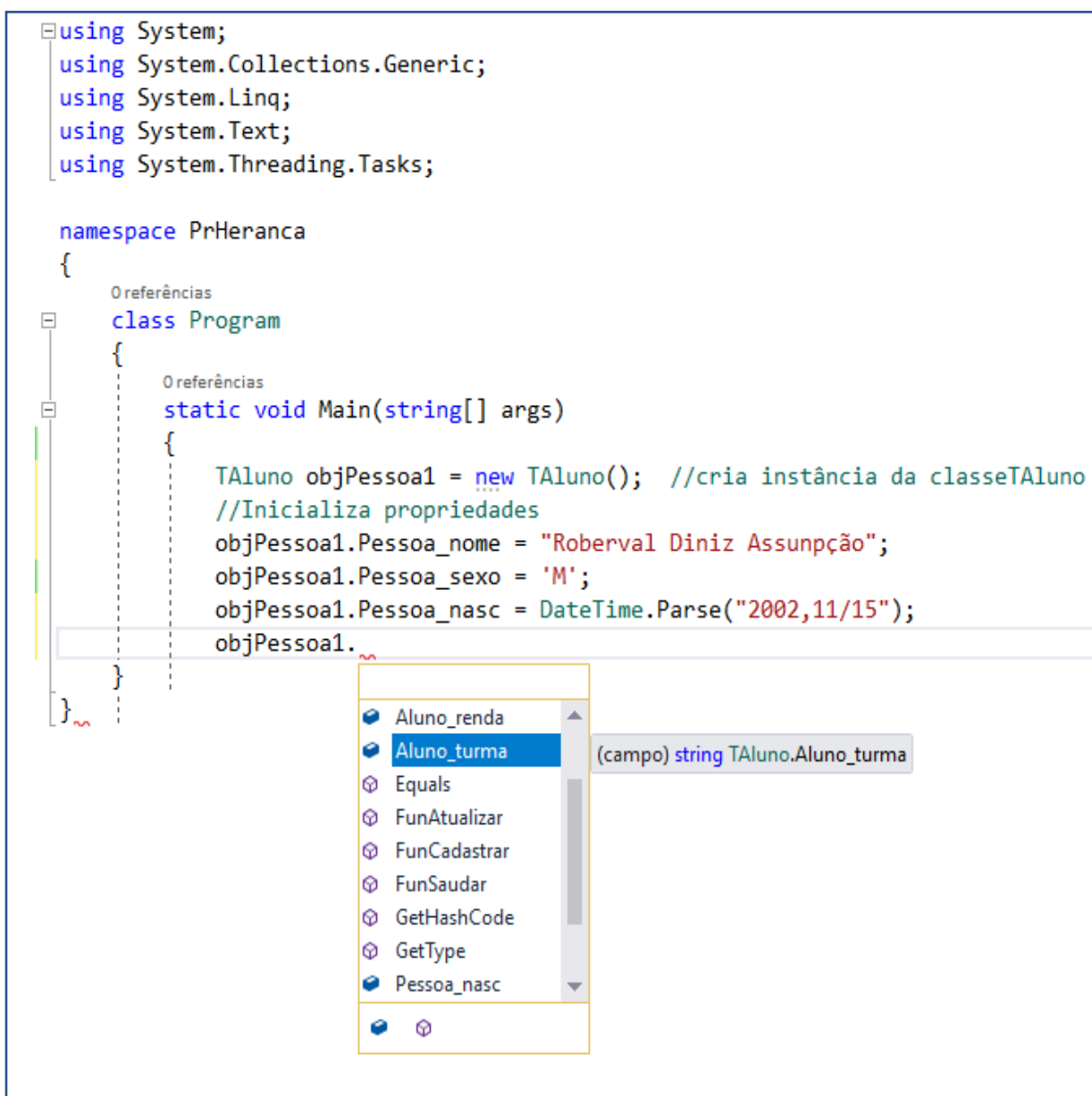


```
D:\Livros\Livro10\Projetos\PrHeranca\PrHeranca\bin\Debug\PrHeranca.exe
Olá; tudo bem!? Meu nome é: Roberval Diniz da Assunção

Roberval Diniz da Assunção
M
15/11/2002 00:00:00
1234
3ADM-N
Roberval Taylor da Assunção
Marta Berenice Assunção
8500,00

Situação do aluno: Aprovado
```

Figura 1 - Saída do programa de herança de classe



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrHeranca
{
    class Program
    {
        static void Main(string[] args)
        {
            TAluno objPessoa1 = new TAluno(); //cria instância da classeTAluno
            //Inicializa propriedades
            objPessoa1.Pessoa_nome = "Roberval Diniz Assunção";
            objPessoa1.Pessoa_sexo = 'M';
            objPessoa1.Pessoa_nasc = DateTime.Parse("2002,11/15");
            objPessoa1.
        }
    }
}
```

Aluno_renda
Aluno_turma (campo) string TAluno.Aluno_turma
Equals
FunAtualizar
FunCadastrar
FunSaudar
GetHashCode
GetType
Pessoa_nasc

Figura 2 - Mostrando os membros da classe derivada TALuno

```

class Program
{
    //Exemplo do mecanismo de Herança
    //Implementado em C#
    //Autor: Mário Leite

    static void Main(string[] args)
    {
        TALuno objPessoa1 = new TALuno(); //cria instância da classe TALuno
        //Inicializa propriedades herdadas
        objPessoa1.Pessoa_nome = "Roberval Diniz da Assunção";
        objPessoa1.Pessoa_sexo = 'M';
        objPessoa1.Pessoa_nasc = DateTime.Parse("2002/11/15");
        //Inicializa as novas propriedades adquiridas
        objPessoa1.Aluno_matr = 1234;
        objPessoa1.Aluno_turma = "3ADM-N";
        objPessoa1.Aluno_pai = "Roberval Taylor da Assunção";
        objPessoa1.Aluno_mae = "Marta Berenice Assunção";
        objPessoa1.Aluno_renda = 8500.00m;

        //Inicializa valores das notas parciais
        double N1 = 7.1;
        double N2 = 7.3;
        double N3 = 6.9;
        double N4 = 6.8;

        //Mostra saudação do aluno
        objPessoa1.FunSaudar(objPessoa1.Pessoa_nome);
        Console.WriteLine();

        //Mostra os dados cadastrais do aluno
        objPessoa1.FunCadastrar(objPessoa1.Pessoa_nome,
                                objPessoa1.Pessoa_sexo, objPessoa1.Pessoa_nasc,
                                objPessoa1.Aluno_matr, objPessoa1.Aluno_turma,
                                objPessoa1.Aluno_pai, objPessoa1.Aluno_mae,
                                objPessoa1.Aluno_renda);
        Console.WriteLine(objPessoa1.Pessoa_nome);
        Console.WriteLine(objPessoa1.Pessoa_sexo);
        Console.WriteLine(objPessoa1.Pessoa_nasc);
        Console.WriteLine(objPessoa1.Aluno_matr);
        Console.WriteLine(objPessoa1.Aluno_turma);
        Console.WriteLine(objPessoa1.Aluno_pai);
        Console.WriteLine(objPessoa1.Aluno_mae);
        Console.WriteLine(objPessoa1.Aluno_renda);
        Console.WriteLine();

        //Mostra o status do aluno
        string situacao = objPessoa1.FunVerificarStatus(N1, N2, N3, N4);
        Console.WriteLine("Situação do aluno: {0}", situacao);
        Console.ReadKey();
    }
} //fim da classe Programa

```

```

class TAluno: TPessoa //cria a classe derivada
{
    //Propriedades adicionais da classe TAluno
    public int Aluno_matr;
    public string Aluno_turma;
    public string Aluno_pai;
    public string Aluno_mae;
    public decimal Aluno_renda;

    //Métodos adicionais da classe TAluno
    public void FunCadastrar(string nome, char sexo,
        DateTime nasc, int matr,
        string turma, string pai, string mae, decimal renda)
    {
        Aluno_matr = matr;
        Aluno_turma = turma;
        Aluno_pai = pai;
        Aluno_mae = mae;
        Aluno_renda = renda;
    }
    public string FunVerificarStatus(double nota1, double nota2,
        double nota3, double nota4)
    {
        double media = (nota1+nota2+nota3+nota4)/4;
        string status;
        if (media >= 7){
            status = "Aprovado";
        }
        else if ((media >= 6.0) && (media < 7.0)) {
            status = "Em Recuperação";
        }
        else if ((media >= 5.0) && (media < 6.0)) {
            status = "Em Exame";
        }
        else {
            status = "Reprovado";
        }

        return status;
    }
} //fim da classe TAluno

```