



# Interface em C#

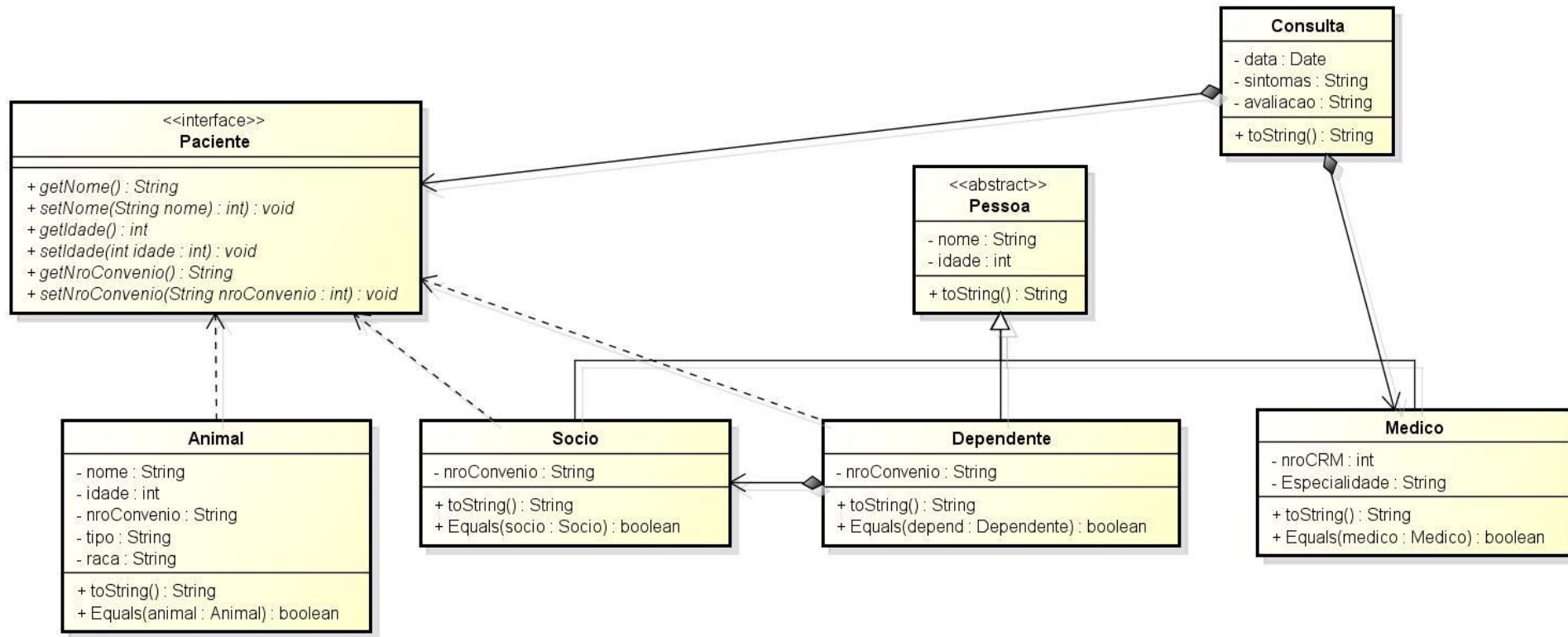
Prof. Gustavo Dibbern Piva



# Interfaces

- ✓ Uma interface é um componente que define padrões de funcionalidade (métodos) e constantes que são implementadas pelas classes que possuem relacionamento com elas.
- ✓ Uma interface possui unicamente assinatura de métodos e constantes.
- ✓ Os métodos constantes de uma interface são abstratos e públicos.
- ✓ Uma classe pode implementar mais de uma interface.

# Exemplo



# Interface Paciente

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

namespace prjExemploInterface
{
    interface Paciente
    {
        String getNome();
        void setNome(String nome);
        int getIdade();
        void setIdade(int idade);
        String getNroConvenio();
        void setNroConvenio(String nroConvenio);
    }
}
```

# Classe Pessoa

```
namespace prjExemploInterface
{
    public abstract class Pessoa
    {
        private String nome;

        public Pessoa()
        {
            this.nome = "";
            this.idade = 0;
            this.cpf = "";
        }

        public Pessoa(String nome, int idade, String cpf)
        {
            this.nome = nome;
            this.idade = idade;
            this.cpf = cpf;
        }

        public String Nome
        {
            get { return nome; }
            set { nome = value; }
        }

        private int idade;

        public int Idade
        {
            get { return idade; }
            set { idade = value; }
        }

        private String cpf;

        public String Cpf
        {
            get { return cpf; }
            set { cpf = value; }
        }
    }
}
```

# Classe Socio

```
namespace prjExemploInterface
{
    public class Socio:Pessoa,Paciente
    {
        private String nroConvenio;

        public String NroConvenio
        {
            get { return nroConvenio; }
            set { nroConvenio = value; }
        }

        public Socio(String nome,int idade,String cpf,String nroConvenio):base(nome,idade,cpf)
        {
            this.nroConvenio = nroConvenio;
        }

        public Socio()
            : base()
        {
            this.nroConvenio = "";
        }

        public String getNome()
        {
            return this.Nome;
        }
        public void setNome(String nome)
        {
            this.Nome = nome;
        }

        public int getIdade()
        {
            return Idade;
        }
        public void setIdade(int Idade)
        {
            this.Idade = Idade;
        }
        public String getNroConvenio()
        {
            return this.nroConvenio;
        }
        public void setNroConvenio(String nroConvenio)
        {
            this.nroConvenio = nroConvenio;
        }
    }
}
```

# Classe Dependente

```
namespace prjExemploInterface
{
    public class Dependente:Pessoa,Paciente
    {
        private String nroConvenio;

        public String NroConvenio
        {
            get { return nroConvenio; }
            set { nroConvenio = value; }
        }

        private Socio socio;

        public Socio Socio
        {
            get { return socio; }
            set { socio = value; }
        }

        public Dependente()
            : base()
        {
            this.nroConvenio = "";
            this.socio = new Socio();
        }

        public Dependente(String nome,int idade,String cpf,String nroConvenio,Socio socio):base(nome,idade,cpf)
        {
            this.nroConvenio = nroConvenio;
            this.socio = Socio;
        }
    }
}
```



# Exercício

- ✓ Implemente as demais classes de acordo com o Diagrama de classes apresentado e crie um formulário para marcar consultas, instanciando pelo menos um animal e um sócio.