



LP2

TEORIA AULA 2

Profª Mª Denilce Veloso
denilce.veloso@fatec.sp.gov.br
denilce@gmail.com

Fatec – ADS – LP2

1

Conceitos Fundamentais de Orientação a Objetos

- Objeto
- Identidade, Atributo, Método
- Classe
- Abstração
- Classificação/Instanciação
- Generalização/Especialização
- Agregação/Decomposição e Composição/Decomposição
- Associação
- Composição
- Encapsulamento
- Herança
- Polimorfismo

Fatec – ADS – LP2

2

Conceitos Fundamentais de Orientação a Objetos – Objeto(Entidade)

Entidade é “algo” ou “alguma coisa”, que existe, possui determinadas características, e desempenha alguma função no Mundo Real.

Conforme o conceito acima uma Entidade também pode ser entendida como um **Objeto**. Veja alguns exemplos.






Fonte imagens: google

Fatec - ADS

3

Conceitos Fundamentais de Orientação a Objetos – Classe (categoria)

Classe de Objetos

“Grupo de objetos com os mesmos atributos e os mesmos comportamentos pertencem a mesma classe.”






Fonte imagens: google

Fatec - ADS

4

Conceitos Fundamentais de Orientação a Objetos – Resolução de problemas através da Orientação a Objetos

A resolução de um problema através do paradigma da Orientação a Objetos inclui:

- ✓ Análise do contexto onde se insere o problema;
- ✓ Definição das entidades ou objetos do problema, e
- ✓ Definição das interações que devem ocorrer entre as entidades;

Exemplos:


- Fazer uma viagem
- Assistir TV
- Sistema Controle de alunos para Academia

Fatec – ADS – LP2

5

Conceitos Fundamentais de Orientação a Objetos – Abstração

Um Modelo de Resolução deve contemplar de forma adequada a situação real (problema do mundo real), e portanto, sua construção deve levar em consideração quais são os aspectos relevantes para a solução do problema, e quais são os irrelevantes. A técnica para análise de uma situação real é denominada **ABSTRAÇÃO**.



“Real” “Virtual”

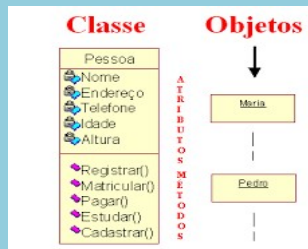
Fatec – ADS – LP2

6

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração

Antes da definição do Modelo de Resolução, é importante ter em mente quais as operações são frequentemente realizadas em um processo de abstração envolvendo objetos.

Exemplo de Classe e Objetos:



7

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração

As operações básicas de um processo de abstração são:

1. Classificação / Instanciação

Nas operações de classificação/instanciação utilizamos a seguinte notação ou representação simbólica, para representar:

A relação é estabelecida usando-se uma linha pontilhada.

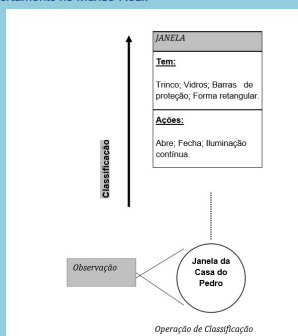


Fatec – ADS – LP2

8

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração – Classificação

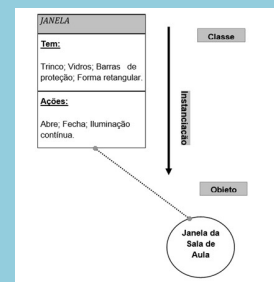
Consiste em definir a Classe ou Categoria a que um objeto pertence, a partir da análise das suas características e comportamento no Mundo Real.



9

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração – Instanciação

Consiste na criação de um OBJETO contendo todas as características e ações de uma determinada CLASSE.



Fatec – ADS – LP2

10

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração – Exercícios

Considerando as seguintes classes, indique os seus atributos e comportamentos (ações) e represente a classificação e a instanciação.

- Empregado
- Conta Corrente
- Aluno

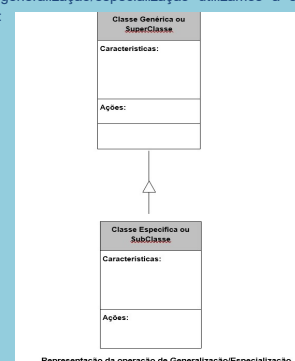
Fatec – ADS – LP2

11

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração

2. Generalização / Especialização

Para representar a operação de generalização/especialização utilizamos a seguinte notação ou representação simbólica:

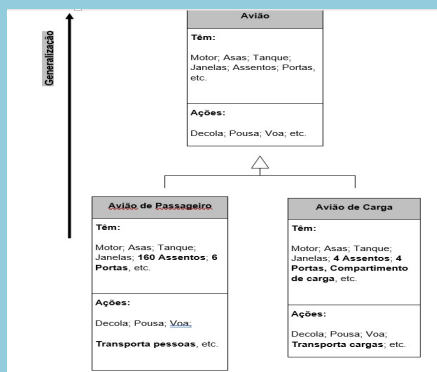


Representação da operação de Generalização/Especialização

12

Conceitos Fundamentais de Orientação a Objetos – Generalização

Operação que consiste na análise de um conjunto de classes, identificando características comuns a todas, com o objetivo de definir uma classe mais genérica que especificará essas características comuns.

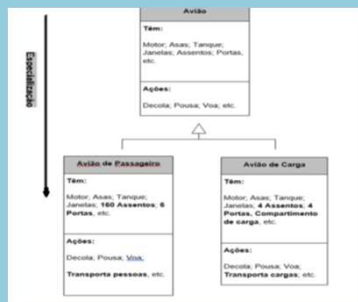


Fatec – ADS – LP2

13

Conceitos Fundamentais de Orientação a Objetos – Especialização

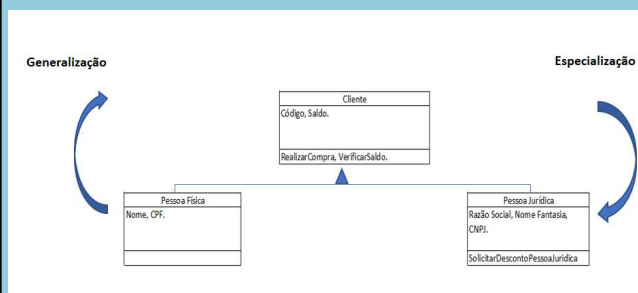
Consiste em identificar uma ou mais SubClasses, a partir de uma Classe mais genérica, especificando características adicionais a essas SubClasses.



Fatec – ADS – LP2

14

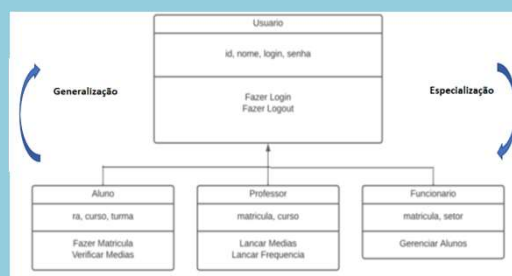
Conceitos Fundamentais de Orientação a Objetos – Generalização / Especialização – Exemplo Conta Cliente (Venda)



Fatec – ADS – LP2

15

Conceitos Fundamentais de Orientação a Objetos – Generalização / Especialização – Exemplo Sistema Escolar



Fatec – ADS – LP2

16

Conceitos Fundamentais de Orientação a Objetos – Generalização / Especialização – Exemplo Animais Vertebrados



Fonte: <https://pt.slideshare.net/rmvandre/animais-vertebrados-paula-pires>

Fatec – ADS – LP2

17

Conceitos Fundamentais de Orientação a Objetos – Generalização / Especialização - Exercícios

Considere as seguintes classes abaixo e represente generalização e especialização

- Conta Bancária
- Aluno Fatec
- Empregado
- Imóvel

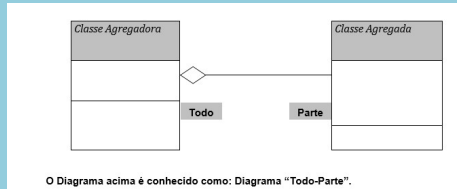
Fatec – ADS – LP2

18

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração

3. Agregação / Decomposição

Para representar a operação de Agregação/Decomposição utilizamos a seguinte notação ou representação simbólica:



Fatec – ADS – LP2

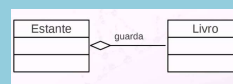
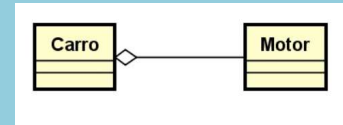
19

19

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração - Agregação

Consiste em definir a composição de um Objeto (Classe).

Quando unimos um conjunto de Objetos, para formar outro Objeto, estamos executando a operação de Agregação.



Fatec – ADS – LP2

20

20

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração - Agregação



Fonte: <https://www.istockphoto.com/br/foto/como-de-carro-desmontado-e-muitas-pe%C3%A7as-de-ve%C3%ADculos-gm927781468-254492081>

Fatec – ADS – LP2

21

21

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração – Decomposição

Consiste na análise de um Objeto (Classe), isolando cada um de seus componentes.



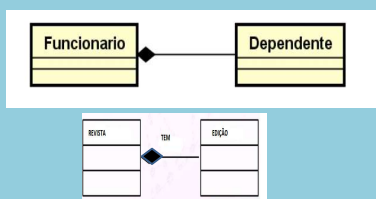
Fatec – ADS – LP2

22

22

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração – Composição (é um tipo de agregação)

Atenção a Composição: também representa uma relação todo-parte entre o agregado e suas parte. Porém a composição representa um forte relacionamento, em que as partes não sobrevivem sem o todo. Sua sintaxe é a de uma linha com um losango preto na classe que corresponde ao todo.



Fatec – ADS – LP2

23

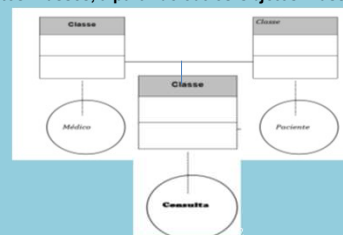
23

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração - Associação

Consiste na identificação de uma ou mais relação entre Objetos.

Na operação de associação, as entidades ou objetos, que se relacionam existem independentemente uma da outra.

A operação de Associação não deve ser confundida com a de Agregação, pois esta última tem por objetivo modelar Objetos/Classes, a partir de outros Objetos/Classes.



24

24

Conceitos Fundamentais de Orientação a Objetos – Operações utilizadas na Abstração - Associação



Fatec – ADS – LP2

25

PILARES DA ORIENTAÇÃO A OBJETOS

ENCAPSULAMENTO: refere-se ao fato de que classes são uma espécie de código completo e fechado; completo no sentido de que define suas próprias variáveis internas (membros de dados), seus eventos, propriedades e métodos que operam sobre os dados internos, e fechado no sentido de que não são permitidas manipulações externas deles – a não ser (e sob condições controladas, definidas pelos projetistas das classes) através de objetos dessa classe.

HERANÇA: é o processo de geração de uma classe a partir de outra existente, disponibilizando à classe gerada os métodos, propriedades e eventos de primeira (chamada classe base), sem necessidade de reescrevê-los.

Fatec – ADS – LP2

26

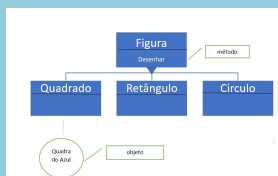
PILARES DA ORIENTAÇÃO A OBJETOS

Sem os dois primeiros pilares não existe o **POLIMORFISMO**, e sem este a POO não seria eficaz.

Polimorfismo significa muitas formas.

Em termos de programação, o **polimorfismo** permite que um único nome de classe ou nome de método represente um código diferente, selecionado por mecanismo automático. Assim, um nome pode assumir muitas formas e como pode representar código diferente, o mesmo nome pode representar muitos comportamentos diferentes.

Exemplo:



Fatec – ADS – LP2

27

Classes no C#

São códigos que definem um tipo de dado, composto de membros de dados internos (variáveis destinadas a armazenamento de valores) e métodos (funções que agem sobre esses membros).

Exemplos: Os tipos de dados no C#.net são implementados como uma classe.

Vejamos o tipo de dado String. Ao definir uma variável de uma classe, você está criando um objeto dela (instanciando a classe).

Exemplo:

String linguagem;

linguagem = "C#.Net";

MessageBox.Show(linguagem.Length.ToString());

MessageBox.Show(linguagem.Substring(0, 3));

Length (é uma propriedade da classe string que retorna o tamanho da string e Substring é um método da classe string que retorna um subconjunto da string).

Nota: Observe que não é necessário você conhecer a estrutura interna da classe string, tudo que você precisa fazer é declarar objetos da classe, quais são as funcionalidades disponíveis e como utilizá-los, é uma espécie de "caixa-preta".

Fatec – ADS – LP2

28

Classes no C#

Exemplo de uma classe criado pelo desenvolvedor:

```
{ class Empregado
{
    private int matricula; // atributo
    private string nomeEmpregado;

    6 referências
    public int Matricula // propriedade
    {
        get { return matricula; }
        set { matricula = value; }
    }

    6 referências
    public string NomeEmpregado
    {
        get { return nomeEmpregado; }
        set { nomeEmpregado = value; }
    }
}
```

Fatec – ADS – LP2

29

Classes e Sub-Classes no C#

Quando você cria uma classe ela herda automaticamente uma classe chamada System.Object. Ela fornece alguns métodos que você pode garantir são suportados por todos os objetos que você terá.

SubClasses

Quando for criar uma subclasse (isto é dependente da superclasse), o procedimento é o mesmo, exceto que deverá ser utilizado o caractere : (dois pontos)

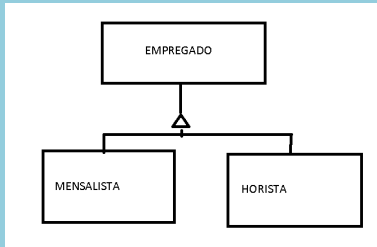
```
class Horista : Empregado
{
    // nao posso herdar mais de uma classe
}
```

Fatec – ADS – LP2

30

➤ Classes no C# - Exercício de Criação de Classes no C#

Criar as seguintes classes no C#, observe que as subclasses Mensalista e Horista são especialização da classe Empregado, portanto irão utilizar herança.



Fatec - ADS - LP2

31

31

➤ Métodos em C#

Lembra do comportamento???
Virou método.

Um método é um bloco de código que contém uma série de instruções. Um programa faz com que as instruções sejam executadas chamando o método e especificando todos os argumentos necessários do método.

Quando usado como o tipo de retorno para um método, void especifica que o método não retorna um valor.

```
public void NomeMetodo() // não tem retorno
{
    // corpo do método
}

public int Quadrado(int i) // tem um retorno do tipo int
{
    // corpo do método
}
```

Fatec - ADS - LP2

32

32

➤ Construtores e instâncias em C#

Construtores de instâncias, ou simplesmente construtores, são métodos chamados automaticamente quando da instanciação de uma classe, os quais implementam as ações necessárias para sua inicialização. Sempre que uma classe é instanciada, um construtor conhecido como "default" é chamado.

Um construtor sempre tem o nome da classe da qual é membro e pode receber parâmetros (construtor customizado). Ele não possui valor de retorno.

Fatec - ADS - LP2

33

33

➤ Construtores e instâncias em C#

Exemplo construtores da classe Depto:

```
class Depto
{
    public int codigo {get; set;}
    public string nomeDepto {get; set;}

    public Depto() // constructor vazio
    {
    }

    public Depto(int codx, string nomex) // constructor recebendo parâmetros
    {
        codigo = codx;
        nomeDepto = nomex;
    }
}
```

Fatec - ADS - LP2

34

34

➤ Propriedade em C#

Uma propriedade é um membro que oferece um mecanismo flexível para ler, gravar ou calcular o valor de um campo particular. As propriedades podem ser usadas como se fossem membros de dados públicos, mas na verdade elas são métodos realmente especiais chamados *acessadores*.

Um acessador de propriedade get é usado para retornar o valor da propriedade e um acessador de propriedade set é usado para atribuir um novo valor.

```
class TimePeriod
{
    private double _seconds;

    public double Seconds
    {
        get { return _seconds; }
        set { _seconds = value; }
    }
}

// ou simplesmente
class TimePeriod
{
    public double Seconds {get; set;}
}
```

Fatec - ADS - LP2

35

35

➤ Escopo dos membros de Dados em C#

A **declaração dos variáveis membros** de uma classe se inicia definindo o escopo dessas variáveis, isto é, onde elas serão visíveis ou não.

public – disponíveis dentro e fora da classe que a declarou (no mesmo projeto ou outros que fazem referência a ele).

Internal – disponíveis não apenas na classe que a declarou, mas em todo o projeto que contém essa classe (não em outros projetos).

private – disponíveis somente dentro da classe que a declarou.

protected – disponíveis não apenas na classe que a declarou, mas em toda a classe derivada dela (através de herança).

protected internal – disponíveis não apenas em todo o projeto que contém a classe que os declara, mas em todas as classes derivadas dela (mesmo que seja em outros projetos/assemblies).

Fatec - ADS - LP2

36

36

➤ Namespaces em C#

Os namespaces são usados intensamente em programas em C# de duas maneiras. Em primeiro lugar, as classes do .NET Framework usam namespaces para organizar suas muitas classes. Em segundo lugar, declarar seus próprios namespaces pode ajudar a controlar o escopo dos nomes de classe e de método em projetos de programação maiores.

37

➤ Static em C#

A palavra-chave **Static** faz com que um método ou propriedade esteja associado a uma classe e não com uma instância particular da classe.

Ele se torna acessível a partir da classe pois não precisa ser acessado através de uma instância da classe. Embora também possa ser acessado desta forma.

Com isto concluímos que membros declarados como **Static** são membros da classe e não membros de instância.

Podemos acessar membros declarados como **Static** em uma classe sem criar uma instância da classe.

Por exemplo:

```
Public static String empresa = "Irmãos Silva Ltda.";
```

38

📖 Referências

- Curso C#. Classes. Disponível em:
https://www.youtube.com/watch?v=tZtYFAy_Js&index=9&list=PLvtPpql8E1jaLoNJ1BUibVfAz0xjnClx Acesso: 16.Ago.2018.
- Coad, Peter. Yourdon, Edward. Análise Baseada em Objetos, Campus, 1992.
- <https://msdn.microsoft.com/pt-br/library/hh972517.aspx> Acesso Jan.2016.
<http://www.macoratti.net/pageview.aspx?catid=18> Acesso Jan.2016.
- Apostilas Prof. Francisco Bianchi curso Delphi - Orientado a Objetos - 2005.

39

Curiosidade – Tecnologias relacionadas

Ver Site

- <https://insights.stackoverflow.com/survey/2020#correlated-technologies>

40