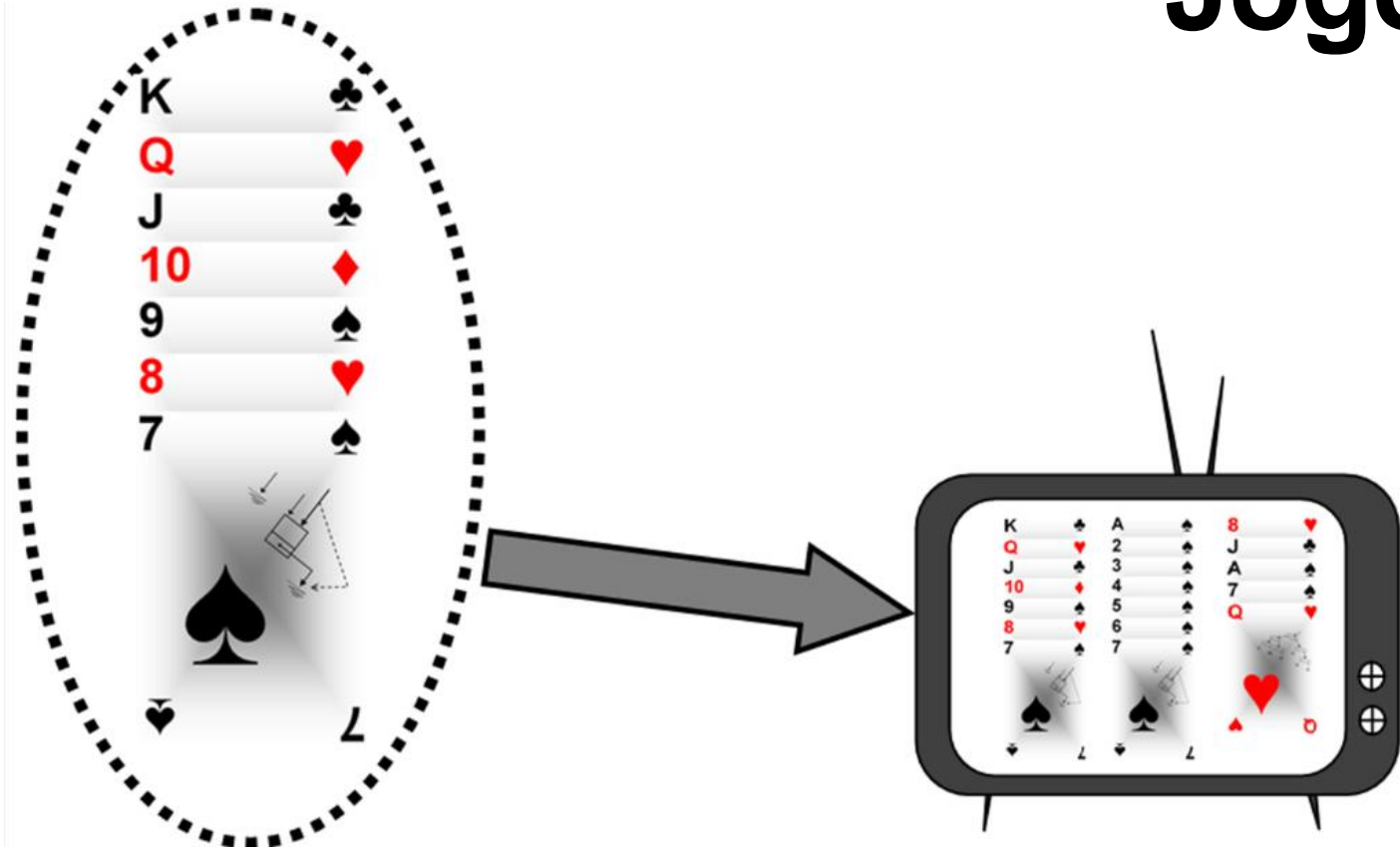


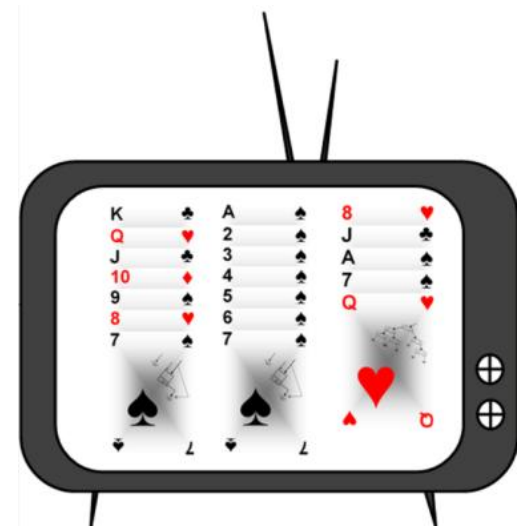
# Estruturas de Dados com Jogos



**Capítulo 1**

**Tipos Abstratos de Dados**

# Seus Objetivos neste Capítulo



- Entender o conceito de Tipos Abstratos de Dados, e o modo de utilizá-lo no desenvolvimento de programas;
- Perceber que o uso de Tipos Abstratos de Dados dá ao software maior portabilidade, maior potencial para reutilização, reduz custos de desenvolvimento e de manutenção;
- Conscientizar-se quanto a importância de adotar uma estratégia que agregue portabilidade e reusabilidade aos jogos que você desenvolverá.

# Fases do Desenvolvimento de Software



The diagram illustrates the phases of software development as a series of five white rectangular boxes with black borders, arranged in a descending staircase pattern from top-left to bottom-right. The boxes are set against a light gray background. The phases are: Análise, Projeto, Implementação, Teste, and Manutenção.

**Análise**

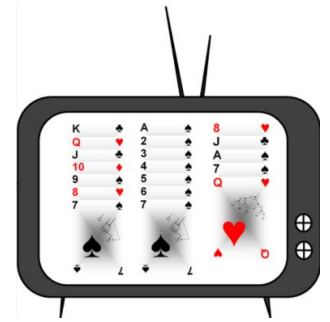
**Projeto**

**Implementação**

**Teste**

**Manutenção**

# Tipo Abstrato de Dados



## Definição - Tipo Abstrato de Dados - TAD:

Um Tipo Abstrato de Dados – TAD - é constituído por um conjunto de **Dados** a serem armazenados, e por um grupo de **Operadores** que podem ser aplicados para manipulação desses Dados.

## Manipulação dos Dados Armazenados:

O armazenamento e a recuperação dos Dados devem ser realizados **exclusivamente** através dos Operadores do TAD.

## Em Que Momento Identificar e Projetar um TAD:

Na fase de Projeto do software.

# Exemplo: *FreeCell*

## TAD Pilha Intermediária do *FreeCell*



### Coleção de Dados a Serem Armazenados

Para cada uma das *Pilhas Intermediárias*:

- As cartas que estão na pilha - valor e naipe de cada carta;
- A sequência das cartas na pilha.

### Operadores Para Manipulação

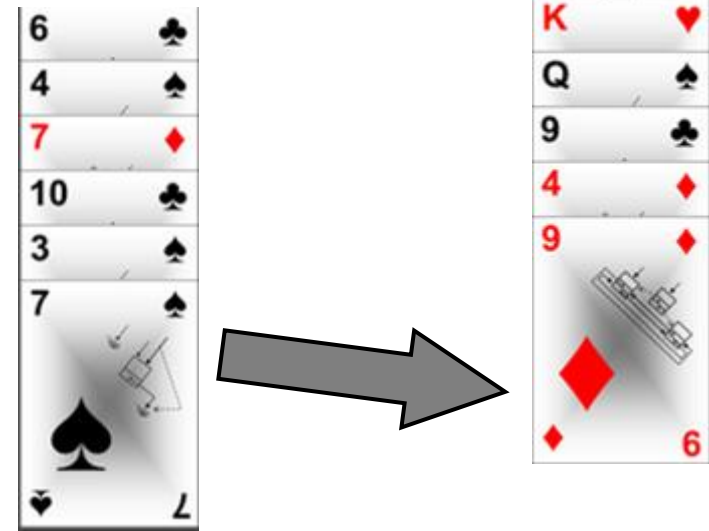
- Retira a carta que está no topo da pilha;
- Coloca uma carta no topo da pilha, se o valor estiver na sequência correta.

# Operações do TAD Pilha Intermediária



- Desempilha (Pilha, Carta, DeuCerto)
- EmpilhaNaSequência (Pilha, Carta, DeuCerto)
- EmpilhaSempre (Pilha, Carta)

# Exercício 1.1 Transfere Carta



**TransfereCarta** (parâmetros por referência **PilhaOrigem**, **PilhaDestino** do tipo **PilhaIntermediária**, parâmetro por referência **DeuCerto** do tipo **Boolean**)

/\* Transfere uma carta da Pilha Origem para a PilhaDestino, caso a carta estiver na sequencia correta na PilhaDestino. O parâmetro DeuCerto retornará o valor Verdadeiro se uma carta for efetivamente transferida, e o valor Falso caso contrário \*/

**TransfereCarta** (parâmetros por referência **PilhaOrigem**, **PilhaDestino** do tipo **PilhaIntermediária**, parâmetro por referência **DeuCerto** do tipo **Boolean**) {

Variável **Carta** do tipo **Carta-do-Baralho**;  
Variável **ConsegiuRetirar** do tipo **Boolean**;  
Variável **ConsegiuEmpilhar** do tipo **Boolean**;

*/\* Tenta retirar Carta do topo da PilhaOrigem \*/*

**Desempilha**(**PilhaOrigem**, **Carta**, **ConsegiuRetirar**);

**Se** (**ConsegiuRetirar** == **Verdadeiro**)

**Então** {

*/\* empilha na Pilha Destino, se estiver na sequência correta \*/*

**EmpilhaNaSequência**(**PilhaDestino**, **Carta**, **ConsegiuEmpilhar**);

**Se** (**ConsegiuEmpilhar** == **Verdadeiro**)

**Então** **DeuCerto** = **Verdadeiro**;

**Senão** { */\* carta não está na sequência correta e deve retornar à*

**PilhaOrigem** \*/

**EmpilhaSempre**(**PilhaOrigem**, **Carta**);

**DeuCerto** = **Falso**;

};

};

**Senão**

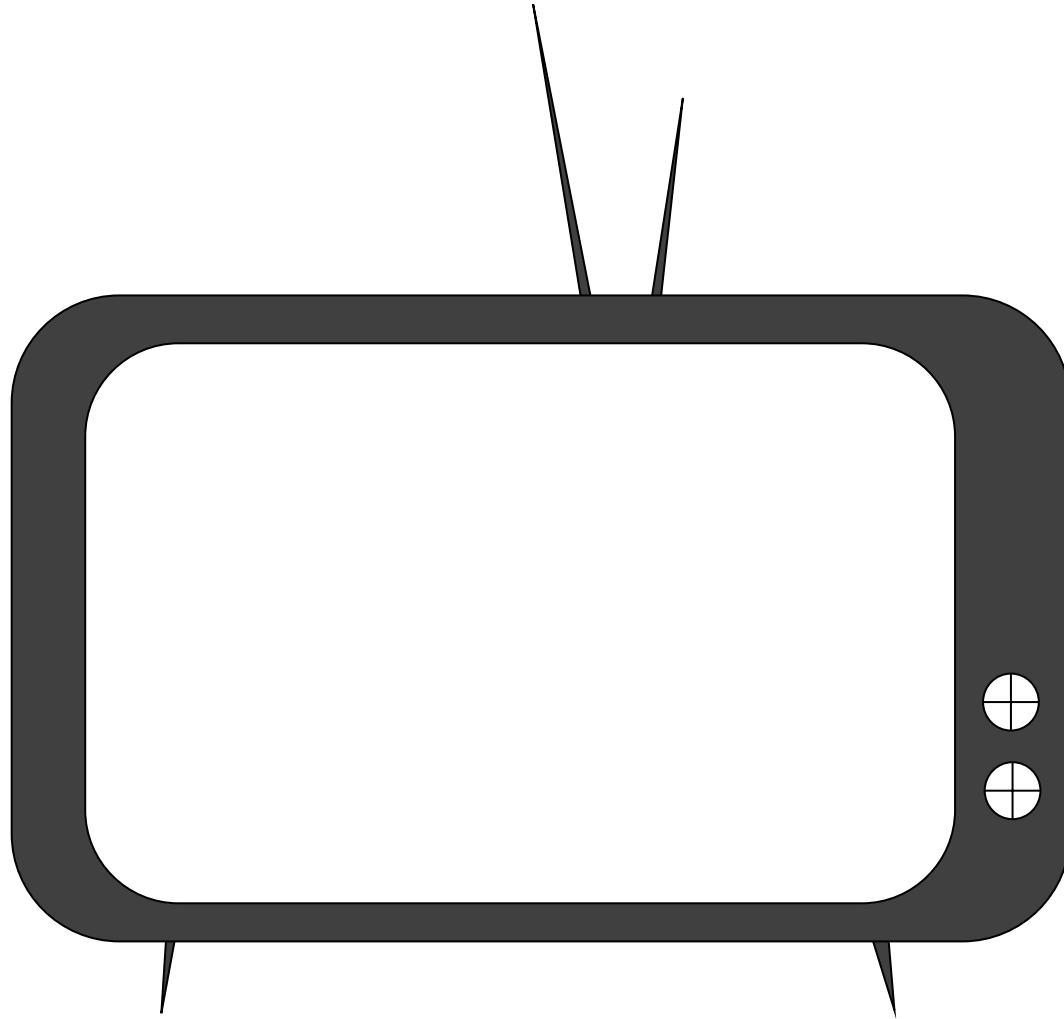
**DeuCerto** = **Falso**;

}


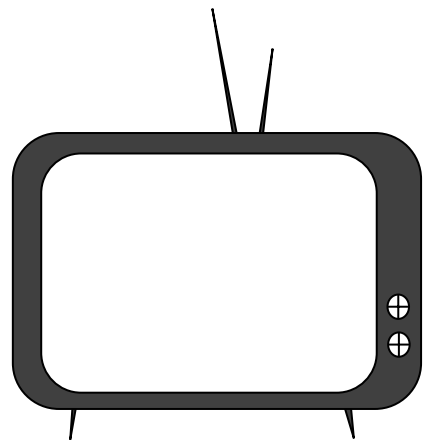




# Qual a Melhor Maneira de Aumentar o Volume da TV?



# Qual a Melhor Maneira de Aumentar o Volume da TV?

	Operadores do TAD ou Botões da TV
	<p><b>Pilha de Cartas:</b></p> <ul style="list-style-type: none"><li>• Retira a carta que está no topo da pilha;</li><li>• Coloca uma carta no topo da pilha, se o valor estiver na sequência correta.</li></ul>
	<p><b>TV:</b></p> <ol style="list-style-type: none"><li>1. Aumenta o volume;</li><li>2. Diminui o volume;</li><li>3. Muda de canal (1 canal acima);</li><li>4. Muda de canal (1 canal abaixo).</li></ol>

# O Que É um Bom Programa?

```
struct Node {
```

```
char Info;
```

```
struct Node *Next; };
```

```
typedef struct Node *NodePtr;
```

```
NodePtr P;
```

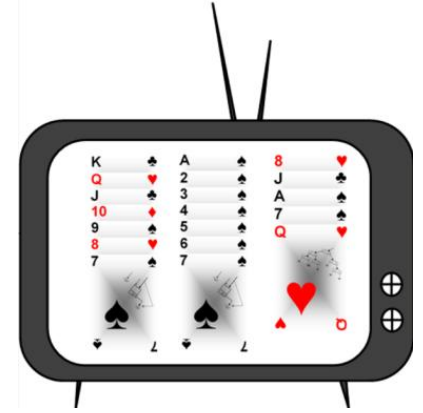
```
P = new Node;
```

# O Que É um Bom Programa?

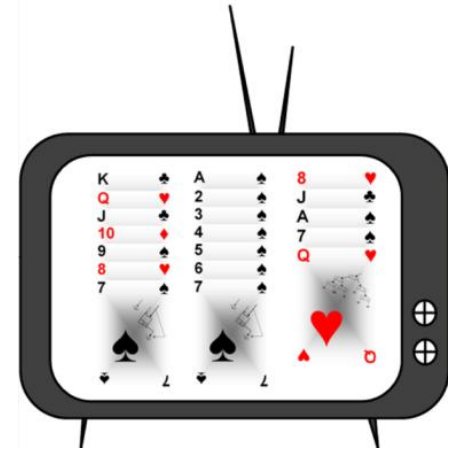
**Portabilidade de Software:** capacidade de executar em diferentes plataformas de hardware e software.

**Reusabilidade de Software:** capacidade de aproveitar (reutilizar) um software já desenvolvido, para satisfazer uma segunda necessidade.

# Vantagens da Utilização de Tipos Abstratos de Dados



- É mais fácil programar, sem se preocupar com detalhes de implantação;
- É mais fácil preservar a integridade dos dados;
- Maior independência e portabilidade de código;
- Maior potencial de reutilização de código.



# Software Bom, Bonito e Barato

O uso do conceito de Tipos Abstratos de Dados aumenta a Portabilidade e o potencial de Reutilização do software. Em consequência disso, o custo de desenvolvimento e manutenção é reduzido.

## Estruturas de Dados com Jogos

Aprender a programar pode ser divertido!