



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 4

Tipos Abstratos de Dados

Separando a Interface da Implementação
TAD

Tipos de Dados

Define o conjunto de **valores** que uma variável pode assumir.

.Tipos primitivos:

- Linguagem C: `char, int, float, double, void`

Ex: `int a`

Domínio de `a` = $\{-2^{31} \text{ a } 2^{31}-1\}$
 $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$

.Tipos definidos pelo programador:

- Utiliza mecanismos de construção de tipos (struct, typedef)

Ex: `typedef struct {
 int dia, mes, ano;
} Data;`

Tipos Abstratos de Dados

Define um conjunto de **valores** e **operações** que podem ser executadas sobre esses valores.

Ex: `int a`

Domínio de `a` = $\{-2^{31} \text{ a } 2^{31}-1\}$



$\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$

Operações = $\{+, -, *, /, \%\}$

- Separa o **conceito** (definição) da **implementação**
(Independência da implementação)

1 . Tipos Abstratos de Dados

Como implementar na Linguagem C?

- 1) Mecanismos de **agrupamento** para definir a forma como os dados serão armazenados:

Exemplo: **Ponto**

(Armazena as coordenadas x,y de um ponto)

```
typedef struct {  
    int x,y;  
} Ponto;
```

- 2) **Funções** para definir as operações:

Exemplo: Operações para Ponto

```
void criaPonto(Ponto *pt, int x, int y);  
void escrevePonto(Ponto pt);  
void movePonto(Ponto *pt, int dx, int dy);  
float calcDistancia(Ponto pt1, Ponto pt2);
```



1 . Tipos Abstratos de Dados

PROBLEMA:

Escreva um programa em C para ler as coordenadas de 2 pontos, calcular e escrever a distância entre eles, mover o segundo ponto de 10 unidades na coordenada x e imprimir o ponto transladado.

```
... /* Os protótipos foram omitidos */

typedef struct {
    int x,y;
} Ponto;

void criaPonto(Ponto *pt,int x,int y) {
    pt->x = x;
    pt->y = y;
}

void escrevePonto(Ponto pt) {
    printf("(%d,%d)", pt.x, pt.y);
}

void movePonto(Ponto *pt, int dx, int dy) {
    pt->x = pt->x + dx;
    pt->y = pt->y + dy;
}

float calcDistancia(Ponto pt1, Ponto pt2) {
    float dist;

    dist = sqrt( pow(pt1.x-pt2.x,2) +
                  pow(pt1.y-pt2.y,2));
    return dist;
}
```

```
int main() {
    Ponto p1,p2;
    int x,y;
    float dist;

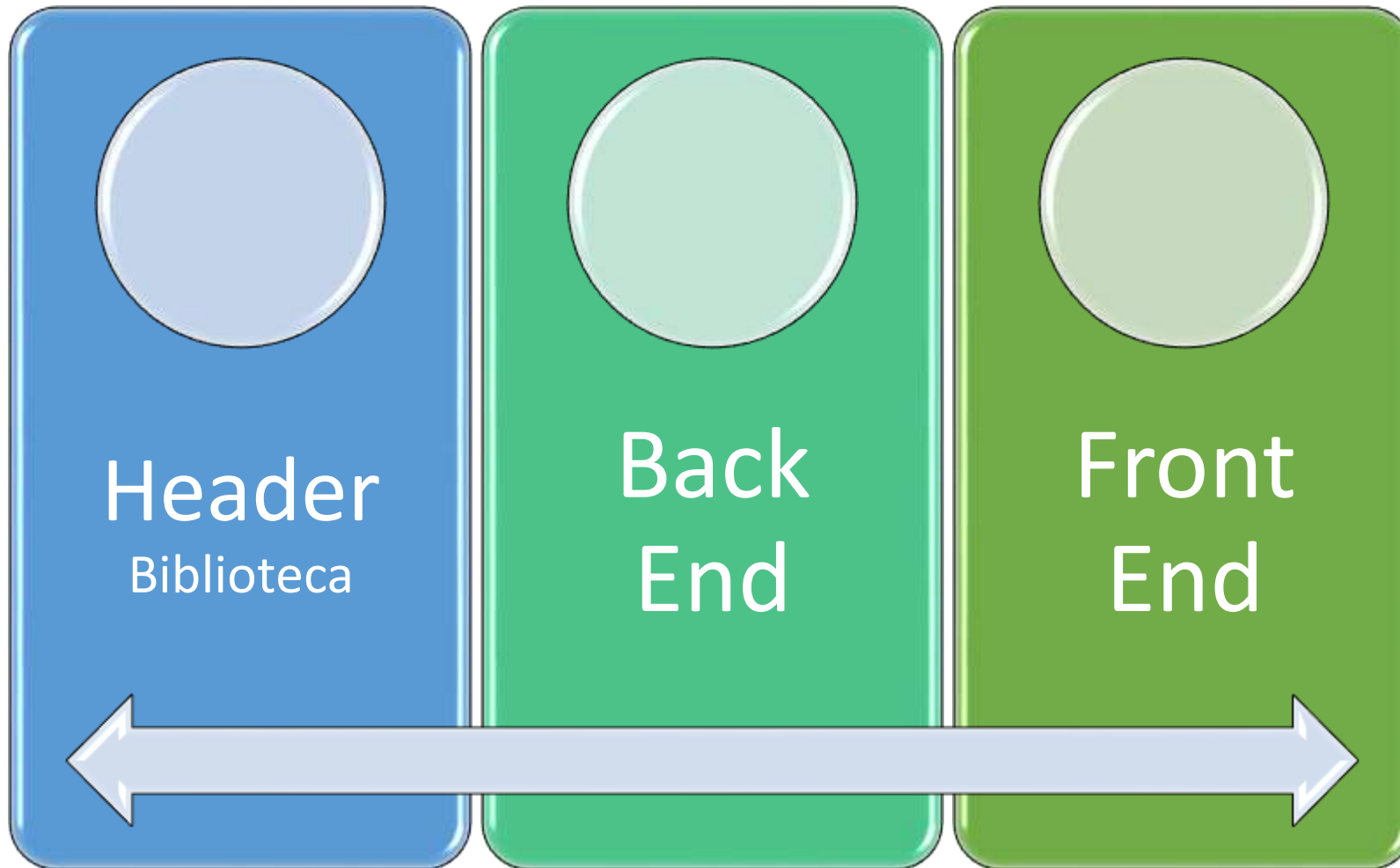
    printf("Informa as coordenadas do ponto 1:");
    scanf("%d %d",&x,&y);
    criaPonto(&p1,x,y);

    printf("Informa as coordenadas do ponto 2:");
    scanf("%d %d",&x,&y);
    criaPonto(&p2,x,y);

    dist = calcDistancia(p1,p2);
    printf("Distancia: %f\n",dist);

    movePonto(&p2,10,0);
    escrevePonto(p2);
    return 0;
}
```

2. Separando a Interface da Implementação



2. Separando a Interface da Implementação

```
/* Arquivo: Circulo.h  
* Contém apenas a declaração do tipo e protótipos  
*/
```

```
#ifndef Circulo_H  
#define Circulo_H
```

Impede múltiplas
inclusões de arquivo
de cabeçalho

```
typedef struct {  
    int x,y;  
    float raio;  
} Circulo;
```

```
void criaCirculo(Circulo *pc,int x,int y,float raio);  
float areaCirculo(Circulo c);  
float perimetroCirculo(Circulo c);
```

```
#endif
```

fim



2. Separando a Interface da Implementação

```
/* Arquivo: Circulo.c
 * Contém a implementação das funções
 */

#include <math.h>
#include "Circulo.h"

void criaCirculo(Circulo *pc,int x,int y,float raio) {
    pc->x = x;
    pc->y = y;
    pc->raio = raio;
}

/*-----*/
float areaCirculo(Circulo c) {
    return M_PI * c.raio * c.raio;
}

/*-----*/
float perimetroCirculo(Circulo c) {
    return 2 * M_PI * c.raio;
}
```



2. Separando a Interface da Implementação

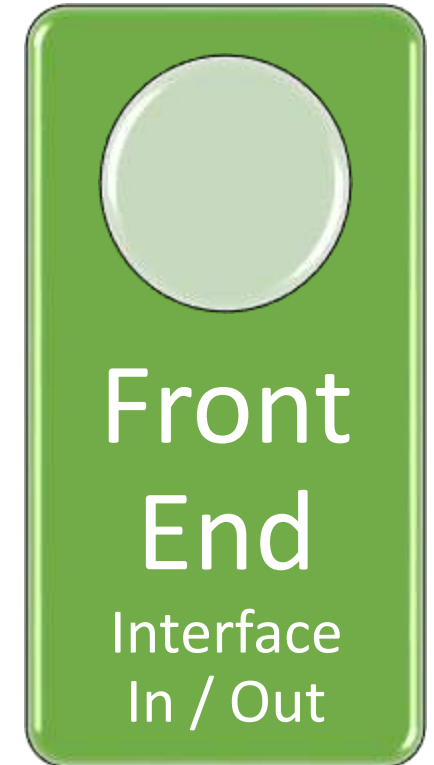
```
/* Arquivo: Exemplo.c
 * Contém a aplicação que usa o tipo Circulo
 */

#include <stdio.h>
#include "Circulo.h"

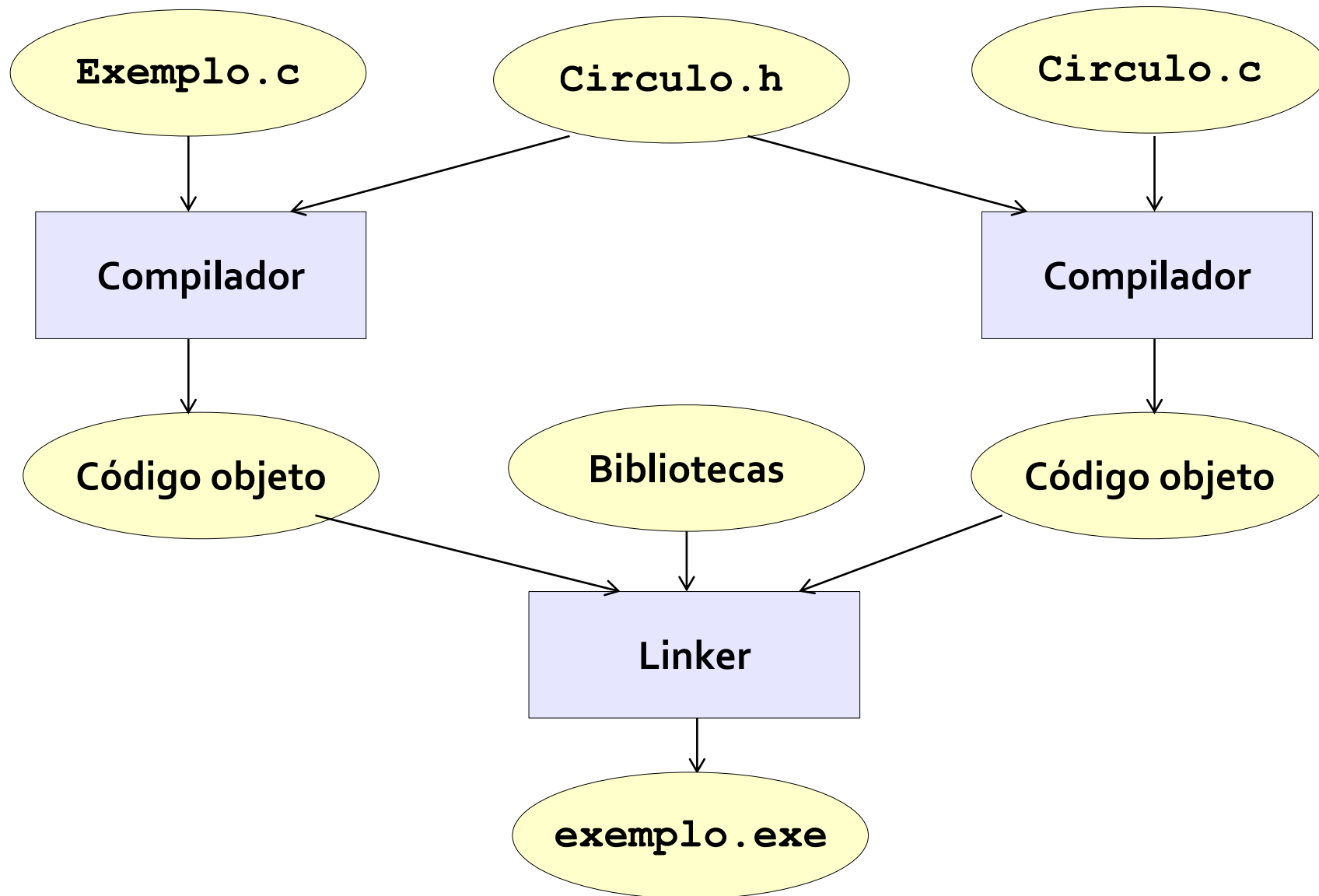
int main() {
    Circulo circ;
    float r;

    printf("Informe o raio:");
    scanf("%f",&r);
    criaCirculo(&circ,0,0,r);

    printf("Area: %f\n", areaCirculo(circ) );
    printf("Perimetro: %f\n", perimetroCirculo(circ) );
    return 0;
}
```

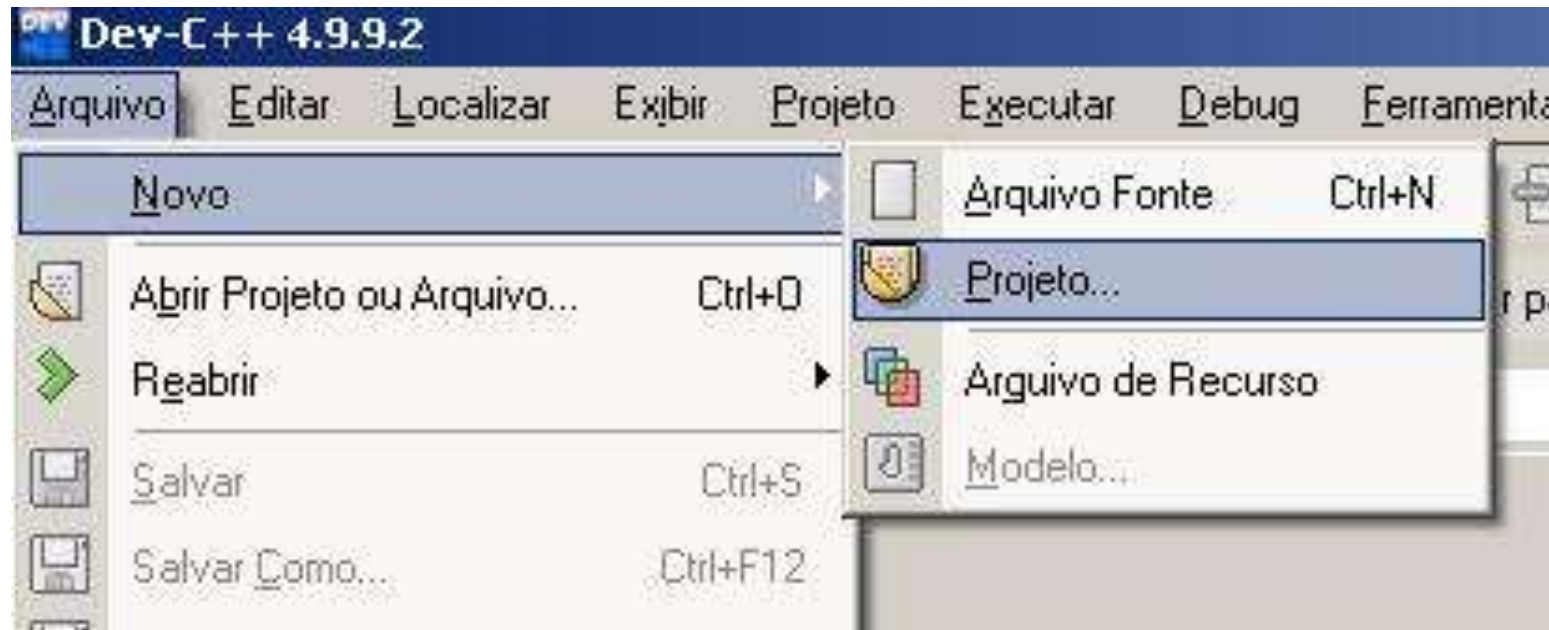


2. Separando a Interface da Implementação



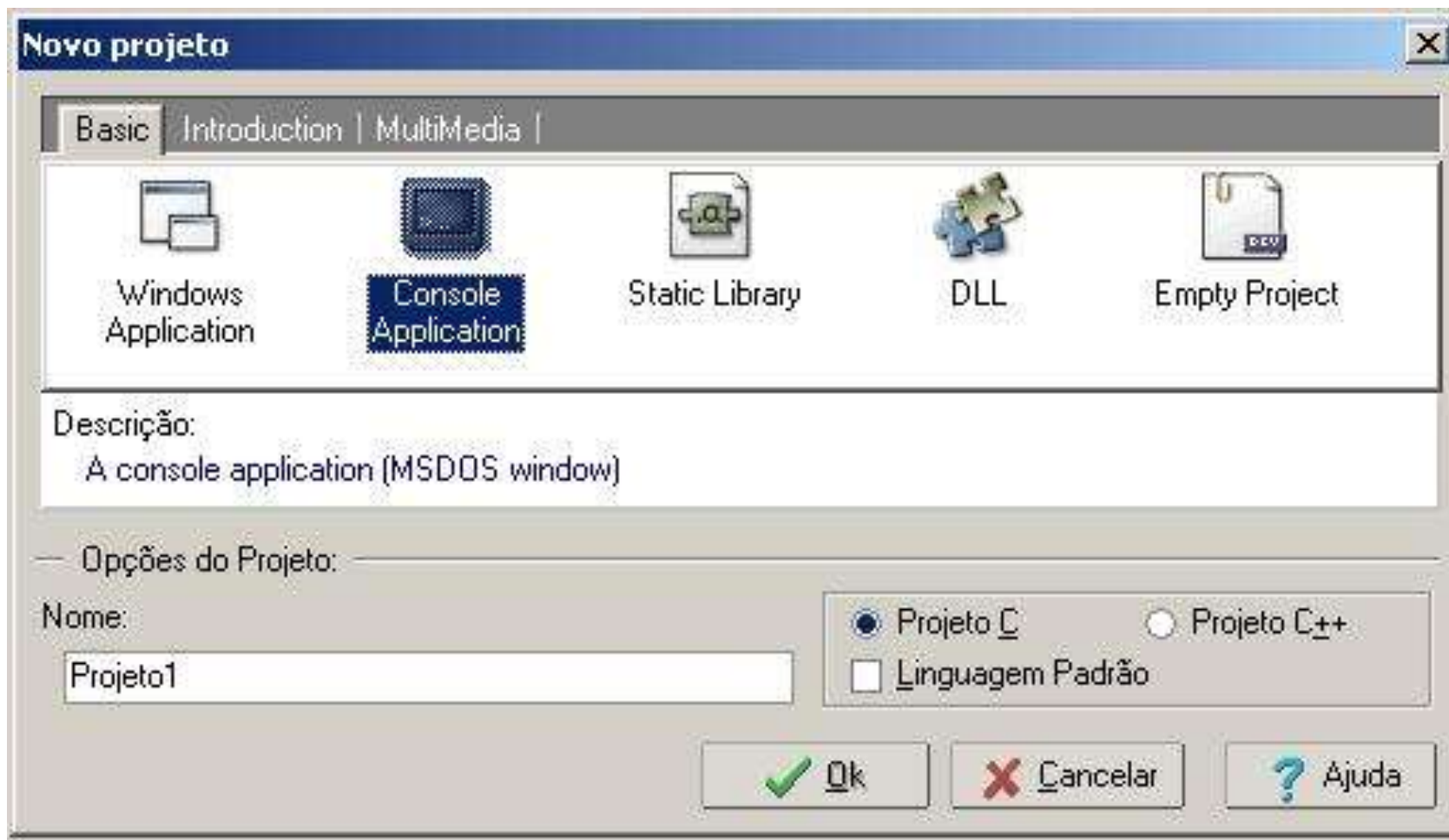
3. Como implementar no DevC++

- Criar um projeto



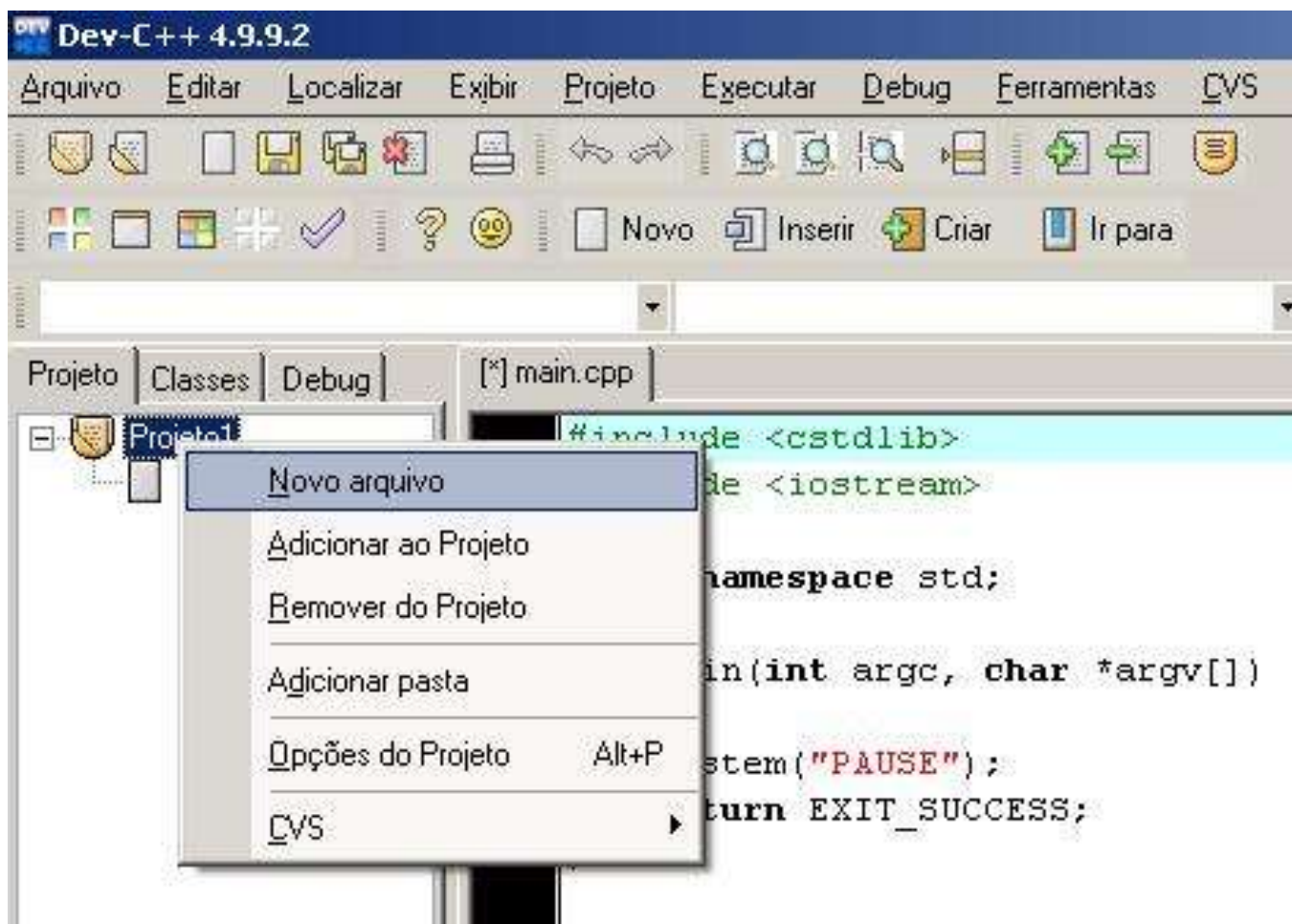
3. Como implementar no DevC++

- Selecionar o tipo de projeto



3. Como implementar no DevC++

- Adicionar um novo arquivo ao projeto





INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 4

Tipos Abstratos de Dados

Separando a Interface da Implementação
TAD