

Estruturas de dados

Separando a interface da implementação

Separando a interface da implementação

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

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

```
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
```

```
/* Impede múltiplas inclusões de
 * arquivo de cabeçalho
 */
```

```
#ifndef Circulo_H
#define Circulo_H
...
#endif
```

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;
}
```

Separando a interface da implementação

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

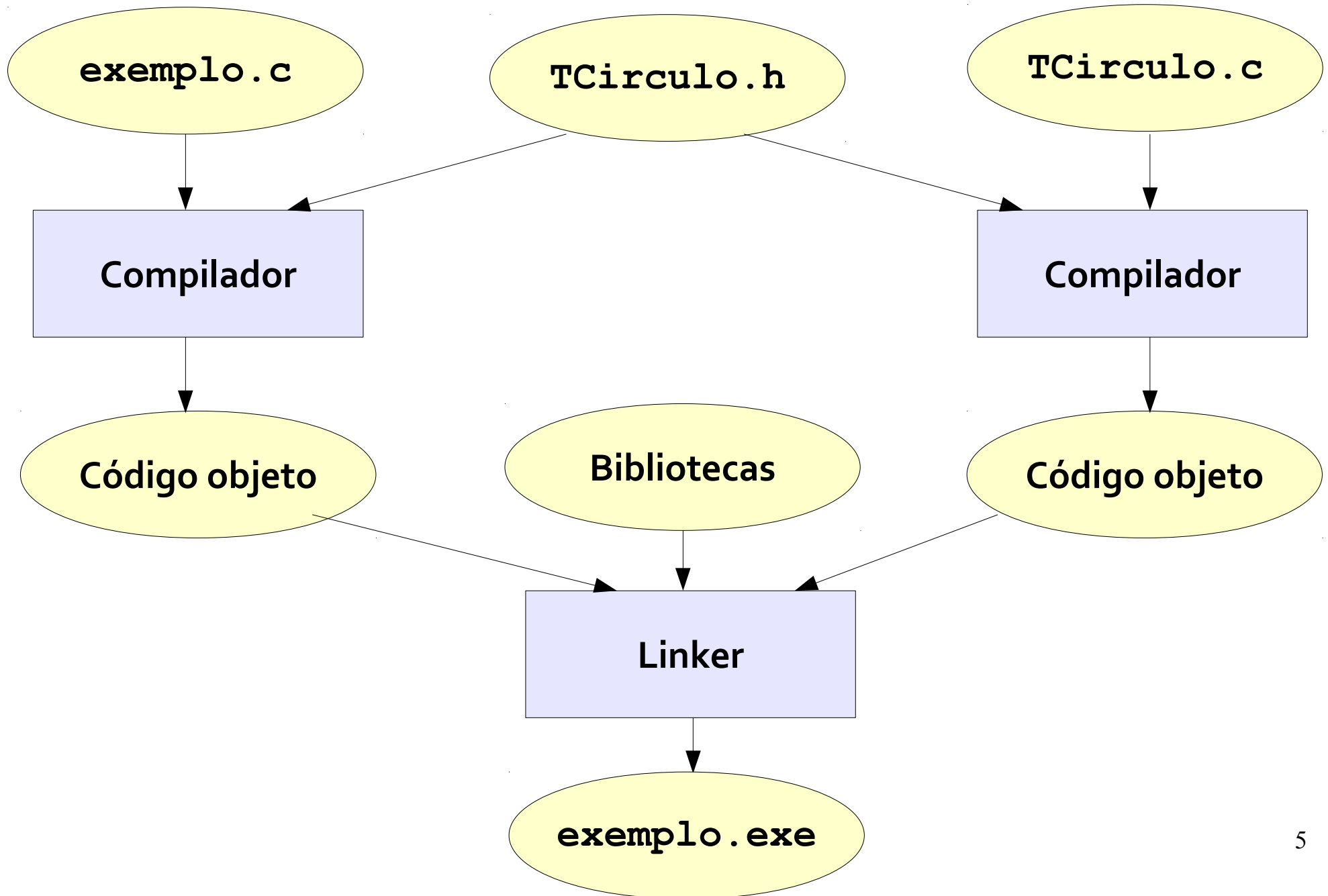
#include <stdio.h>
#include "Circulo.h"

int main() {
    Circulo circ;
    float r;

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

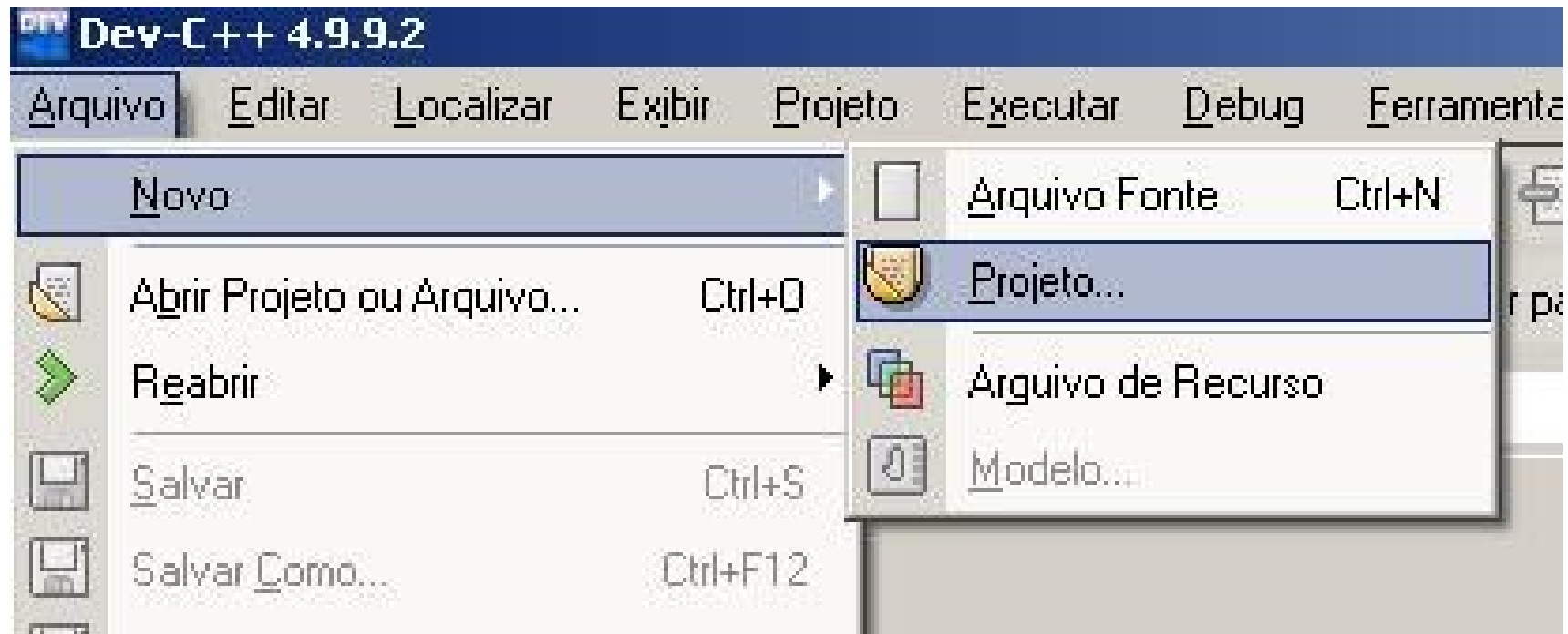
    printf("Area: %f\n", areaCirculo(circ) );
    printf("Perimetro: %f\n", perimetroCirculo(circ) );
    return 0;
}
```

Separando a interface da implementação



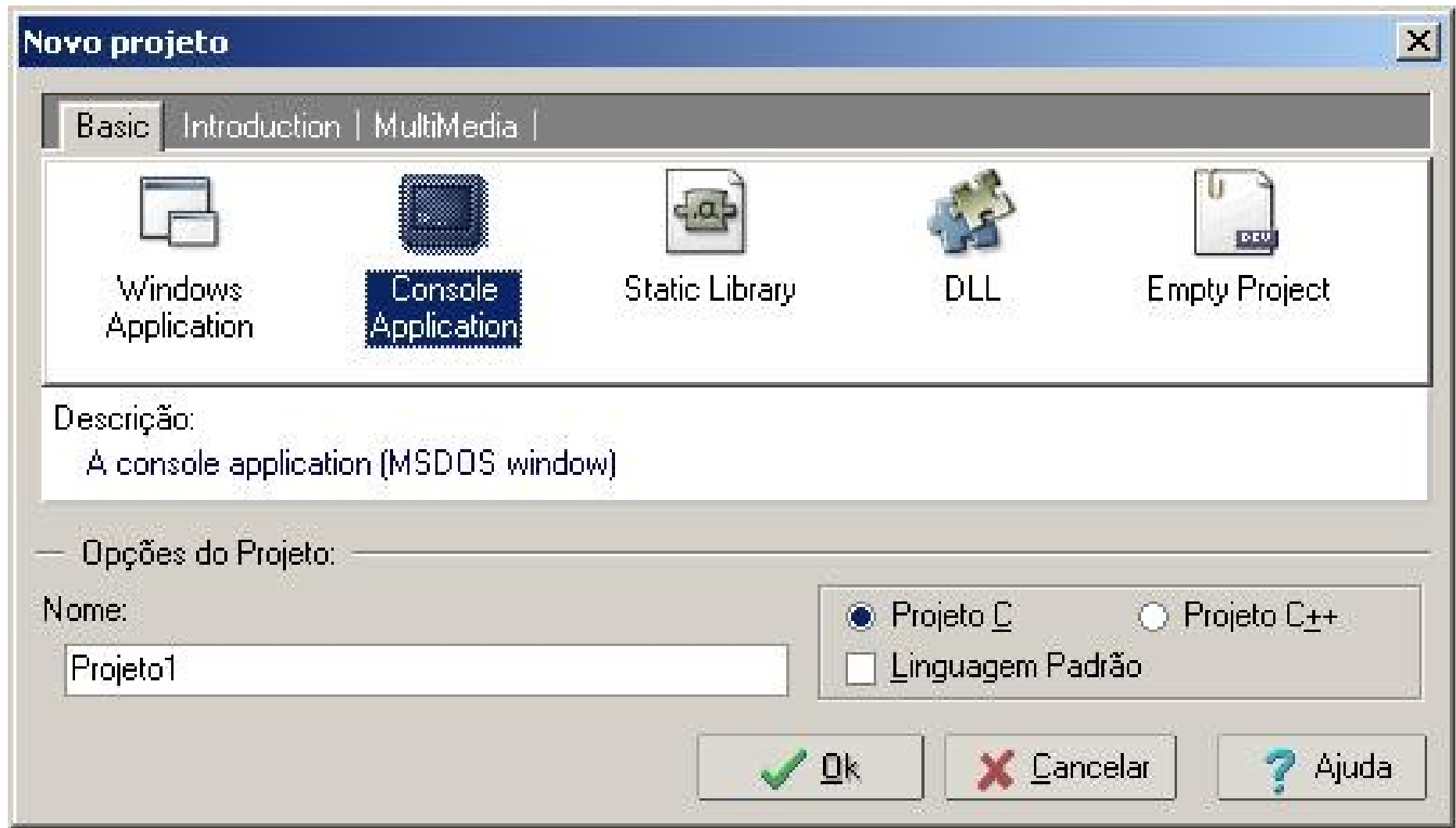
Como implementar no DevC++

- Criar um projeto.



Como implementar no DevC++

- Selecionar o tipo de projeto.



Como implementar no DevC++

- Adicionar um novo arquivo ao projeto

