

TUTORIAL

Como criar arquivos separados em um projeto no Code::Blocks

Prof. Nelio Alves - nelioalves.com

Introdução

No estudo de modularização nas aulas de Lógica de Programação, apresentamos os exemplos em aula de modo que todo o código de um programa ficasse em um único arquivo, para simplificar e agilizar as aulas.

Entretanto, em programas maiores, é altamente recomendável dividir o código em vários arquivos, deixando um arquivo praticamente dedicado a conter a função main, sendo que as demais partes do código fiquem em outros arquivos, de modo que o projeto fique mais organizado.

Assim, vamos refazer o Exercício 13 da Lista 6 de modo que os códigos dos tipos e funções fiquem em um arquivo separado. Vamos lá!

O Exercício 13

13) Fazer um programa para ler os dados de um financiamento de imóvel (valor do imóvel, valor de entrada, número de prestações financiadas). Em seguida, mostrar na tela o valor de cada prestação. Use um tipo estruturado para abstrair o conceito de um financiamento.

Solução em um único arquivo

```
#include <stdio.h>

//-----
// DECLARACAO DE TIPOS

typedef struct {
    float valorImovel;
    float valorEntrada;
    int numeroPrestacoes;
} Financiamento;

//-----
// ASSINATURAS DAS FUNCOES

float valorDaPrestacao(Financiamento);

//-----
// PROGRAMA PRINCIPAL

int main(void) {
    Financiamento x;
    float v;

    printf("Digite os dados do financiamento:\n");
    printf("Valor do imovel: ");
    scanf("%f", &x.valorImovel);
    printf("Valor da entrada: ");
    scanf("%f", &x.valorEntrada);
    printf("Numero de prestacoes: ");
```

```

scanf("%d", &x.numeroPrestacoes);

v = valorDaPrestacao(x);
printf("O valor da prestacao eh: R$ %.2f\n\n", v);

return 0;
}

//-----
// IMPLEMENTACAO DAS FUNCOES

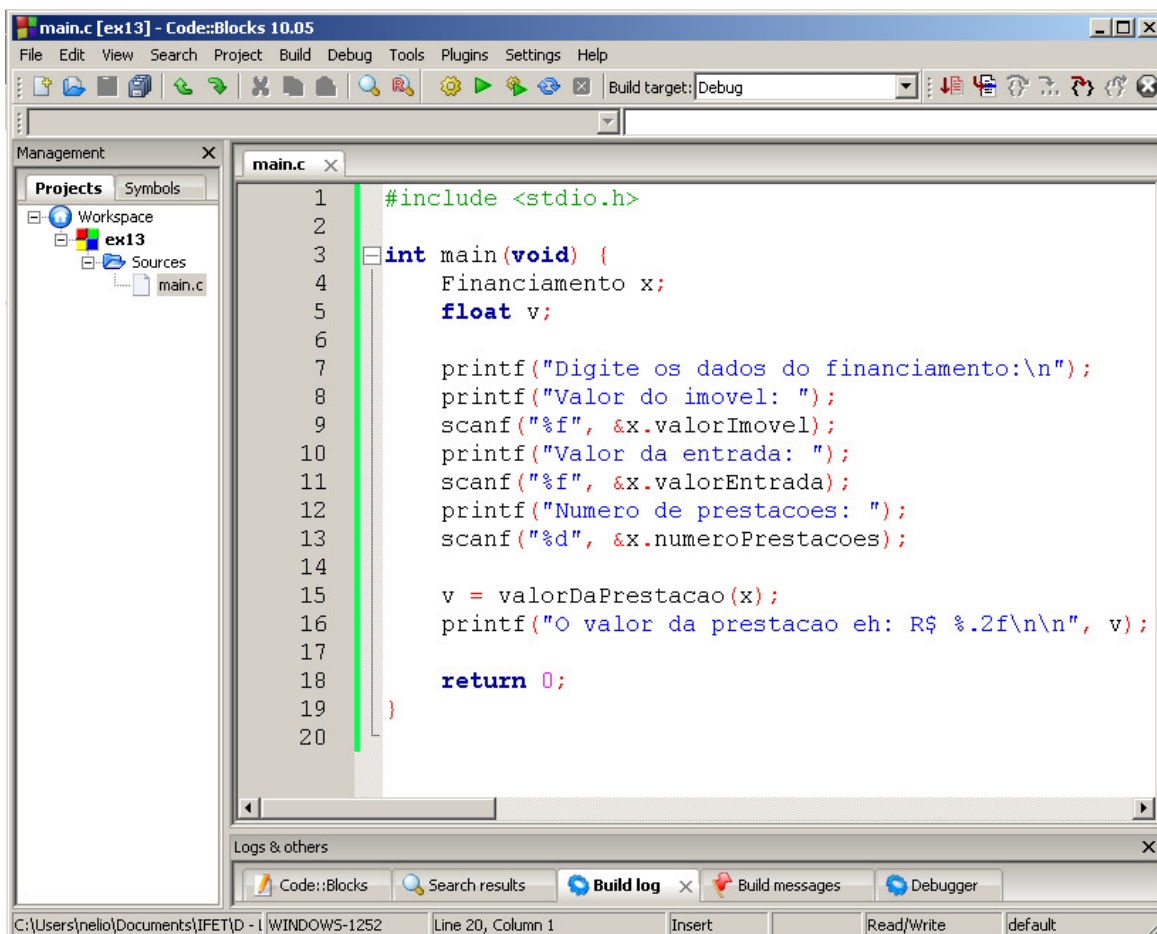
//-----
// RECEBE: um financiamento
// EFEITO: retorna o valor da prestacao do financiamento
float valorDaPrestacao(Financiamento f) {
    float aux;
    aux = (f.valorImovel - f.valorEntrada) / f.numeroPrestacoes;
    return aux;
}

```

Solução usando arquivos separados (passo a passo)

1) Crie um novo projeto no Code::Blocks

2) No arquivo main.c que o Code::Blocks cria automaticamente, coloque o código da função main:



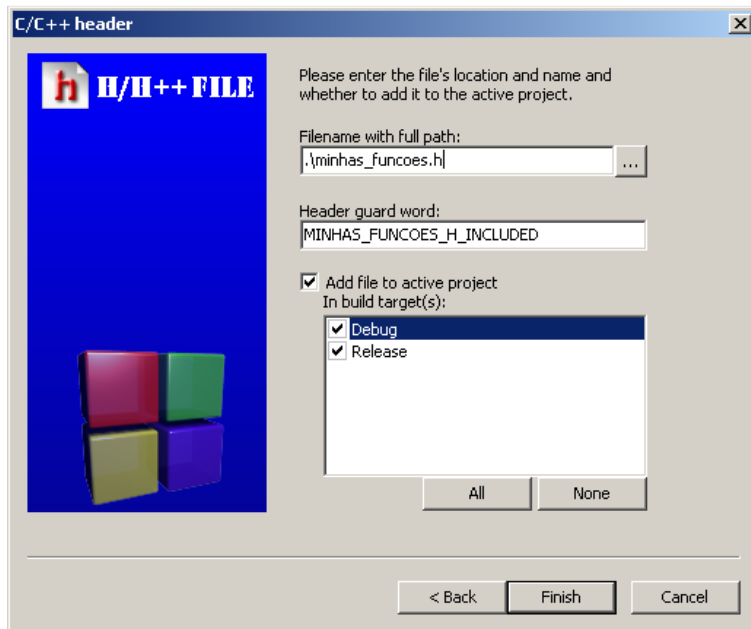
3) Crie um arquivo de cabeçalho (arquivo .h):

3.1) File -> New -> File

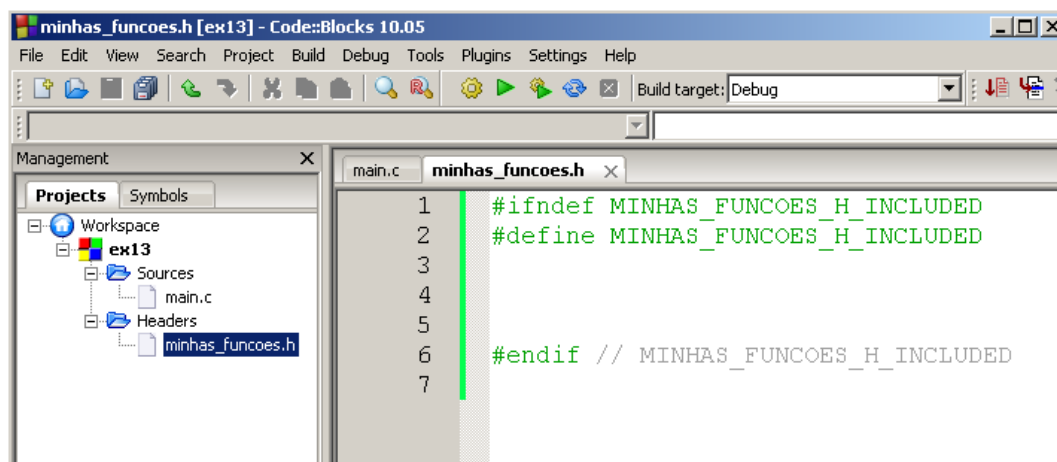
3.2) Escolha "C/C++ Header" e clique em "Go"

3.3) Clique em "Next"

3.4) Preencha o formulário como mostrado abaixo (veja que escolhi o nome "**minhas_funcoes.h**" para meu arquivo e o "." indica que o arquivo ficará na mesma pasta do projeto) e clique em "Finish":

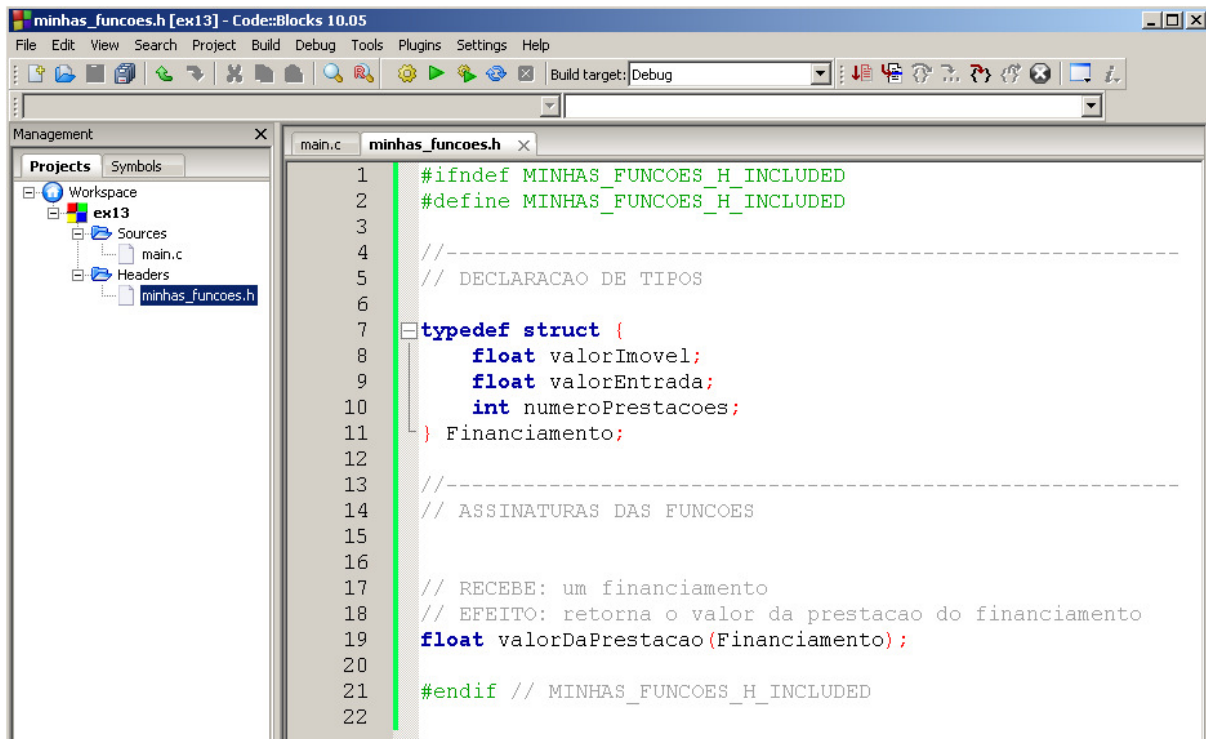


3.5) Confira o resultado. Deverá ter aparecido mais um arquivo na árvore do seu projeto:



OBS.: as cláusulas `#ifndef`, `#define` e `#endif` servem para auxiliar o compilador na hora de importar as bibliotecas. Qualquer dúvida procure o professor.

4) Coloque a declaração de tipo e a assinatura da função (já comentada) no arquivo "**minhas_funcoes.h**" que você acabou de criar, como mostrado a seguir:



5) Crie um arquivo “.c” com o **MESMO NOME** do arquivo “.h” que você criou.

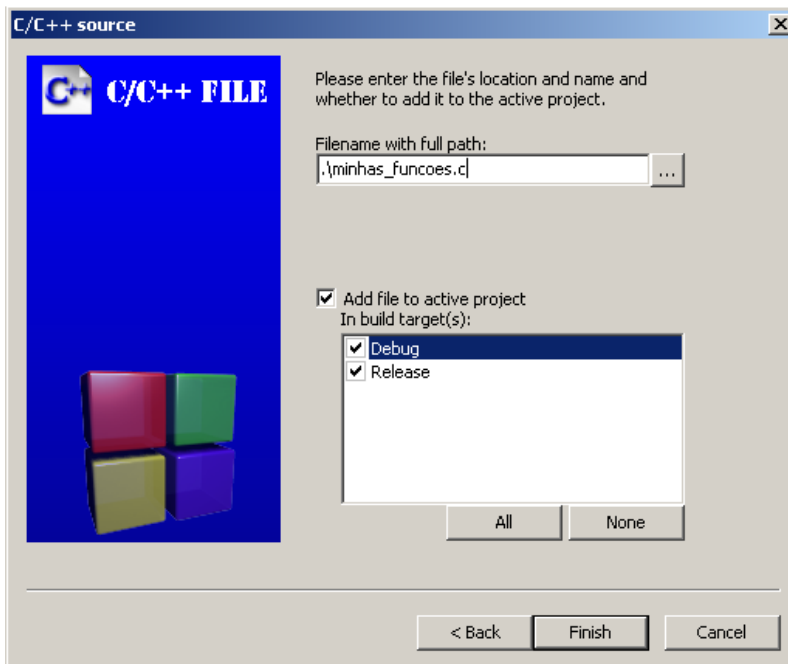
5.1) File -> New -> File

5.2) Escolha “C/C++ Source” e clique em “Go”

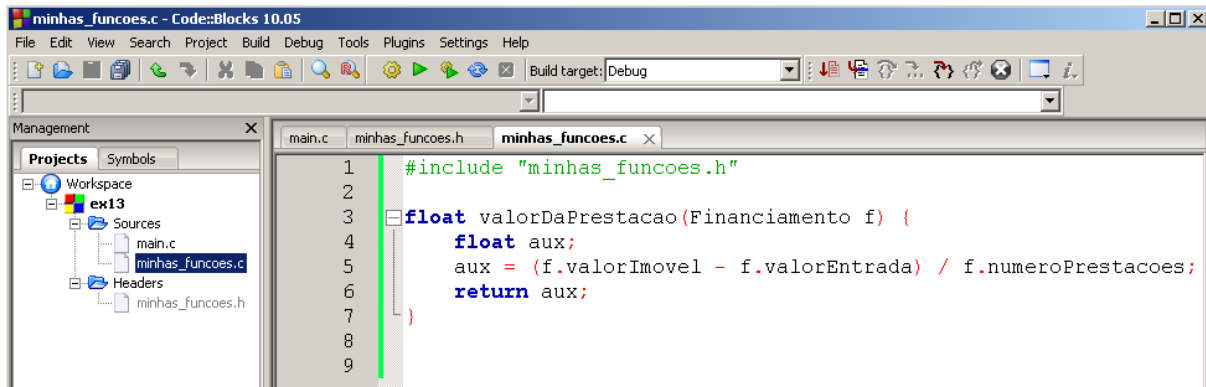
5.3) Clique em “Next”

5.4) Escolha a linguagem “C” e clique em “Next”

5.5) Preencha o formulário como mostrado abaixo. Depois clique em “Finish”:



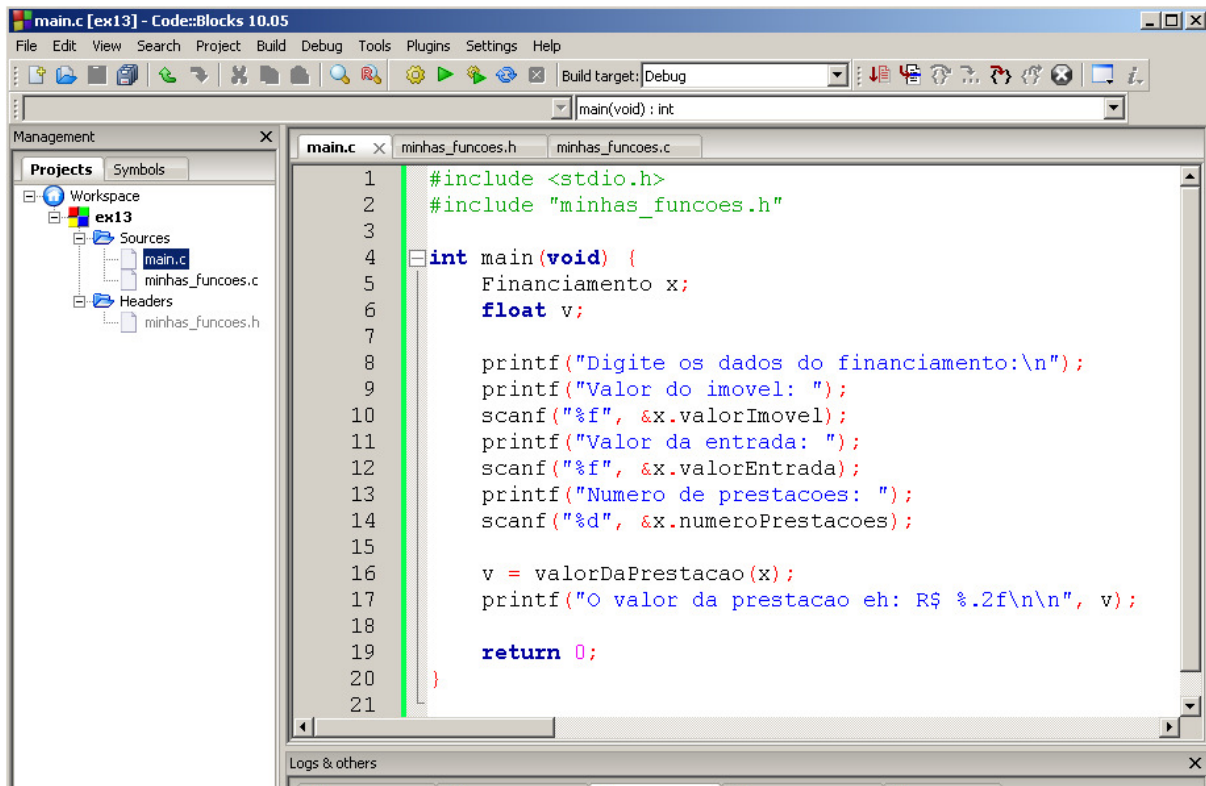
6) Coloque a implementação da função no arquivo “.c” que você acabou de criar (não esqueça de incluir o arquivo “.h” correspondente), como mostrado abaixo:



The screenshot shows the Code::Blocks 10.05 IDE with the 'minhas_funcoes.c' file open. The left sidebar shows the project structure with 'minhas_funcoes.c' selected under 'Sources'. The main editor displays the following C code:

```
1 #include "minhas_funcoes.h"
2
3 float valorDaPrestacao(Financiamento f) {
4     float aux;
5     aux = (f.valorImovel - f.valorEntrada) / f.numeroPrestacoes;
6     return aux;
7 }
8
9
```

7) Inclua o arquivo “.h” também no arquivo “main.c”:



The screenshot shows the Code::Blocks 10.05 IDE with the 'main.c' file open. The left sidebar shows the project structure with 'main.c' selected under 'Sources'. The main editor displays the following C code:

```
1 #include <stdio.h>
2 #include "minhas_funcoes.h"
3
4 int main(void) {
5     Financiamento x;
6     float v;
7
8     printf("Digite os dados do financiamento:\n");
9     printf("Valor do imovel: ");
10    scanf("%f", &x.valorImovel);
11    printf("Valor da entrada: ");
12    scanf("%f", &x.valorEntrada);
13    printf("Numero de prestacoes: ");
14    scanf("%d", &x.numeroPrestacoes);
15
16    v = valorDaPrestacao(x);
17    printf("O valor da prestacao eh: R$ %.2f\n\n", v);
18
19    return 0;
20 }
21
```

PRONTO! Basta executar o programa agora.