



C语言算法设计教程

C语言基础

1. 变量与数据类型

在C语言中，变量是存储数据的容器，而数据类型定义了变量可以存储的数据种类和范围。C语言支持多种基本数据类型，包括整型（int）、字符型（char）、浮点型（float、double）和布尔型（通过#include <stdbool.h>引入的bool）。

1.1 示例：使用不同数据类型的变量

```
#include <stdio.h>

int main() {
    // 定义不同数据类型的变量
    int age = 25;           // 整型变量
    char grade = 'A';      // 字符型变量
    float height = 1.75;   // 浮点型变量
    bool is_student = true; // 布尔型变量，需要引入<stdbool.h>

    // 输出变量的值
    printf("年龄: %d\n", age);
    printf("成绩: %c\n", grade);
    printf("身高: %.2f 米\n", height);
    printf("是否是学生: %s\n", is_student ? "是" : "不是");

    return 0;
}
```

2. 控制结构

控制结构允许程序根据条件执行不同的代码路径，或重复执行一段代码。C语言中的控制结构主要包括条件语句（if、else、switch）和循环语句（for、while、do-while）。

2.1 示例：使用条件和循环控制结构

```
#include <stdio.h>

int main() {
    int number = 10;
```

```

// 使用if-else条件语句
if (number > 0) {
    printf("这是一个正数。\\n");
} else if (number < 0) {
    printf("这是一个负数。\\n");
} else {
    printf("这是零。\\n");
}

// 使用for循环
for (int i = 0; i < 5; i++) {
    printf("循环次数: %d\\n", i);
}

// 使用while循环
int i = 0;
while (i < 5) {
    printf("使用while循环: %d\\n", i);
    i++;
}

return 0;
}

```

3. 函数与模块化

函数是C语言中用于实现特定功能的代码块，可以接受参数并返回值。模块化编程通过将程序分解为多个函数来提高代码的可读性和可维护性。

3.1 示例：定义和调用函数

```

#include <stdio.h>

// 定义一个函数，计算两个整数的和
int add(int a, int b) {
    return a + b;
}

int main() {
    int x = 5;
    int y = 3;
    int result = add(x, y); // 调用add函数
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/218031065024006111>