

一、C语言程序基本结构

- **头文件**：以“#include”开头，用于包含相关的头文件，如“#include <stdio.h>”，引入标准输入输出函数的声明，使程序能够使用printf、scanf等函数。
- **主函数**：C语言程序必须有且仅有一个main函数，它是程序的入口点，程序总是从main函数开始执行。例如“int main(void)”，void表示函数无返回值，也可以写成“int main()”，默认返回值类型为int。
- **语句结束符**：每个语句都以分号“;”结尾，分号是语句的一部分。
- **注释**：用于帮助读者理解程序，不被编译。单行注释以“//”开头，多行注释用“/*”开始，“*/”结束。
- **函数体**：由大括号“{ }”括起来，包含了程序要执行的具体代码，大括号还常用于复合语句、循环语句和选择语句中。

二、数据类型

- **整型**：包括char（通常占1字节）、short（通常占2字节）、int（通常在32位和64位系统上分别占4字节和8字节）、long（在32位系统上通常占4字节，64位系统上占8字节）、long long（通常占8字节）。
- **浮点型**：float（单精度浮点数，通常占4字节）、double（双精度浮点数，通常占8字节）、long double（扩展精度浮点数，占用字节数较多）。
- **布尔型**：bool，需包含头文件<stdbool.h>，占1字节，值为true或false。

三、常量与变量

- **常量**：程序执行期间值不改变的数据。包括整型常量（如十进制123、八进制017、十六进制0x1A）、实型常量（如3.14、2.0e - 5）、字符常量（用单引号括起来，如'A'，还有转移字符如'\n'表示换行）、字符串常量（用双引号括起来，如"Hello, World!"）和符号常量（通过#define定义，如#define PI 3.14）。
- **变量**：用于存储数据的标识符，使用前需声明，如“int age;”。变量可在声明时初始化，如“int count = 0;”，变量命名需以字母或下划线开头，由字母、数字、下划线组成。

四、运算符与表达式

- **算术运算符**：包括+（加法）、-（减法）、*（乘法）、/（除法）、%（取模）。例如“int a = 5, b = 3; int result = a + b;”，result的值为8。
- **关系运算符**：==（等于）、!=（不等于）、<（小于）、>（大于）、<=（小于等于）、>=（大于等于），用于比较操作，结果为真或假（在C语言中用非0表示真，0表示假）。
- **逻辑运算符**：&&（逻辑与）、||（逻辑或）、!（逻辑非）。例如“(a > 3) && (b < 5)”，只有当a > 3和b < 5都为真时，整个表达式才为真。

- **赋值运算符**：=，用于给变量赋值，如“x = 10;”，还存在复合赋值运算符，如+=、-=等，“x += 5;”等价于“x = x + 5;”。

五、流程控制语句

- **if语句**：基本结构为“if (条件表达式) {语句块1;} else {语句块2;}”，当条件表达式为真时，执行语句块1，否则执行语句块2，else部分可省略。
- **switch语句**：用于多分支选择，基本结构为“switch (表达式) {case 常量表达式1: 语句块1; break; case 常量表达式2: 语句块2; break; ... default: 语句块n;}”，根据表达式的值与各个case后的常量表达式进行匹配，执行相应的语句块，break用于跳出switch语句。
- **for循环**：格式为“for (初始化表达式; 条件表达式; 更新表达式) {语句块;}”，先执行初始化表达式，然后判断条件表达式，为真则执行语句块，再执行更新表达式，重复此过程直到条件表达式为假。
- **while循环**：“while (条件表达式) {语句块;}”，只要条件表达式为真，就会一直执行语句块。
- **do - while循环**：“do {语句块;} while (条件表达式);”，先执行一次语句块，再判断条件表达式，为真则继续执行语句块，否则结束循环。
- **break和continue**：break用于跳出当前循环或switch语句，continue用于结束本次循环，直接进入下一次循环的条件判断。

六、函数

- **函数定义**：由返回类型、函数名、参数列表和函数体组成。例如“int add(int a, int b) {return a + b;}”，返回类型为int，函数名是add，参数为a和b，函数体实现两数相加并返回结果。
- **函数声明**：在使用函数之前，通常需要先声明函数原型，告知编译器函数的返回类型、函数名和参数类型，如“int add(int a, int b);”，函数声明末尾需加分号。

七、数组

- **一维数组**：定义形式为“数据类型 数组名[数组大小];”，如“int arr[5];”，可在定义时初始化，如“int arr[5] = {1, 2, 3, 4, 5};”，通过下标访问数组元素，下标从0开始。
- **二维数组**：定义形式为“数据类型 数组名[行数][列数];”，如“int matrix[3][4];”，可用于存储表格形式的数据。

八、指针

- 指针是一种变量，用于存储内存地址。定义指针变量的形式为“数据类型 *指针变量名;”，如“int p;”表示定义一个指向int类型数据的指针p。通过“&”运算符可以获取变量的地址，如“p = &a;”将变量a的地址赋给指针p，通过“*”运算符可以访问指针所指向的内存单元的值，如“int b = *p;”将指针p所指向的值赋给变量b。

九、结构体, 枚举, 联合体, 共用体

- 结构体是一种自定义数据类型，用于将不同类型的数据组合在一起。定义形式为“struct 结构体名 {数据类型 成员名1; 数据类型 成员名2;...};”，如“struct Student {int id; char name[20]; float score;}”，可通过“.”运算符访问结构体变量的成员，如“struct Student stu; stu.id = 1;”。

十、文件操作

- **文件打开**：使用fopen函数，如“FILE *fp = fopen("test.txt", "r");”以只读模式打开名为test.txt的文件，若打开成功返回文件指针，否则返回NULL。
- **文件读取**：可使用fscanf、fgets等函数从文件中读取数据，如“fscanf(fp, "%d", &num);”从文件中读取一个整数到变量num中。
- **文件写入**：使用fprintf、fputs等函数向文件中写入数据，如“fprintf(fp, "%d", num);”将变量num的值写入文件中。
- **文件关闭**：使用fclose函数关闭文件，如“fclose(fp);”。

© 本文章内部分资源来源于网络或 AI 生成，侵权联系删除
本文章仅支持个人学习使用，不允许商用