

C 语言知识整理

2019年C语言程序设计课程讲义资料

Author: Del Cooper

Email: ismdeep@protonmail.com

第1章 程序设计基础知识

1. 什么是计算机语言?
2. 什么是程序?
3. 什么是程序设计?
4. 什么是算法?
5. 流程图
6. 结构化程序设计方法
 - 三种基本结构
 - 顺序
 - 选择
 - 循环
7. 程序设计的风格

第2章 C语言概述

1. C语言基本结构

```
#include <stdio.h>

int main()
{
    return 0;
}
```

2. 预处理指令

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>
```

3. 变量声明

```
int a;
```

4. 函数声明

```
int max(int x, int y)
{
    ....
}
```

5. 变量名、函数名等标识符命名规则

- 由字母（包括大写字母、小写字母）、下划线和数字组成。其中标识符的第一个字符不可以为数字。
- 标识符命名区分大小写。例如：`age` 和 `Age` 为两个不同的标识符。
- 标识符不可以与保留字同名：例如：`int` 是C语言整数类型的类型名称，那么在定义标识符时就不可以再使用 `int` 作为标识符。

6. 关键字（也叫保留字）

关键字（也叫保留字）是C语言预定义好的字符序列，是C语言一个重要的组成部分。

C语言常用的关键字如下：

```
auto break case char const continue default do double else
float for goto while inline int long register return short
signed static struct switch typedef union unsigned void
```

7. ...

第3章 数据类型与运算规则

重点：数据类型；十进制、二进制、八进制、十六进制互相转换；二进制原码、反码、补码；自增、自减

第4章 顺序结构程序设计

1. 基本输入输出语句

- 输入

格式： `scanf("格式化输入控制符", &变量名);`

样例： `scanf("%d", &n);`

- 输出

格式： `printf("格式化输出控制符", 常量、变量或表达式);`

样例： `printf("%d", n);`

- 格式化输入输出控制符

格式字符	对应数据类型	含义
<code>%d</code>	<code>int</code>	接受整数值并将它表示为有符号的十进制整数
<code>%5d</code>	<code>int</code>	举例为 <code>%5d</code> ，输出整数占5位，不够五位就在前面补空格。
<code>%05d</code>	<code>int</code>	举例为 <code>%05d</code> ，输出整数占5位，不够五位就在前面补 0。
<code>%u</code>	<code>unsigned int</code>	无符号10进制整数
<code>%o</code>	<code>unsigned int</code>	无符号的8进制整数（不输出前缀0）
<code>%x</code>	<code>unsigned int</code>	无符号16进制整数
<code>%ld</code>	<code>long</code>	十进制的长整数
<code>%f</code>	<code>float</code>	小数点形式表示的单精度浮点数
<code>%.2f</code>	<code>float</code>	输出小数点后保留2位
<code>%5.2f</code>	<code>float</code>	举例为 <code>%5.2f</code> 表示输出一共占5位，其中小数点后保留2位
<code>%lf</code>	<code>double</code>	小数点形式表示的双精度浮点数
<code>%.2lf</code>	<code>double</code>	输出小数点后保留2位
<code>%5.2lf</code>	<code>double</code>	举例为 <code>%5.2lf</code> 表示输出一共占5位，其中小数点后保留2位
<code>%e</code>	<code>double</code>	科学计数法表示的数
<code>%c</code>	<code>char</code>	输出一个字符
<code>%s</code>		输出一个字符串

2. ...

第5章 选择结构程序设计

01 选择结构

1. if 结构

```
if (表达式)
{
    语句块
}
```

2. if - else 结构

```
if (表达式)
{
    语句块1
}
else
{
    语句块2
}
```

3. if - else if - else 结构

```
if (表达式1)
{
    语句块1
}
else if (表达式2)
{
    语句块2
}
else if (表达式3)
{
    语句块3
}
else
{
    语句块n
}
```

4. switch - case 结构

```
switch(表达式)
{
    case E1:
        语句组1; break;
    case E2:
        语句组2; break;
    ...
    case En:
        语句组n; break;
    default:
        语句组; break;
}
```

第6章 循环结构程序设计

02 循环结构

1. while 结构

```
while (表达式)
{
    语句块
}
```

2. do - while 结构

```
do
{
    语句块
} while (表达式);
```

3. for 结构

```
for (初始; 条件; 自增)
{
    语句块(循环体)
}
```

03 循环嵌套

```
while (条件1)
{
    while (条件2)
    {
        语句块
    }
    语句块
}
```

```
while (条件1)
{
    do
    {
        语句块
    } while (条件2);
    语句块
}
```

```
for (初始; 条件; 自增)
{
    for (初始2; 条件2; 自增2)
    {
        ....
    }
    ....
}
```

04 continue 和 break

1. `continue;`
跳回到最近的循环的自增
2. `break;`
结束最近的循环

第7章 数组

7.1 一维数组

1. 一维数组的定义

类型 数组名[常量表达式];

例如: `int a[100];`

2. 一维数组取值和赋值

`c = a[0];` 表示取出 `a` 数组中下标为 `0` 的数值赋值给变量 `c`。

`a[0] = c;` 表示取出变量 `c` 中的值赋值给 `a` 数组中下标为 `0` 的单元。

注意：C 语言中数组的下标是从 `0` 开始的。也就是说，如果一个数组声明为：`int a[5];` 那么这里面可以使用的下标就为：`0, 1, 2, 3, 4`

3. 一维数组的初始化

```
int a[5] = {1,2,3,4,5};
```

4. 一维数组的应用之寻找最大值

```
include <stdio.h>

int main()
{
    int n, i, m;
    int a[100];

    /* 输入n, 表示一共有n个整数。 */
    scanf("%d", &n);

    /* 接下来输入n个整数到a数组中去。 */
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    /* 假设 a[0] 为数组中最大的。 */
    m = 0;

    /* 依次拿数组中值和假设的最大值位置比较, 如果找到更大的, 则替换假设最大值位置。 */
    for (i = 1; i < n; i++)
    {
        if (a[i] > a[m])
        {
            m = i;
        }
    }

    /* 输出最大值位置和最大值。 */
    printf("a[%d] = %d\n", m, a[m]);
    return 0;
}
```

5. 一维数组的应用之选择排序

```
#include <stdio.h>
```

```

int main()
{
    int n, i, m, t, left;
    int a[100];

    /* 输入n, 表示一共有n个整数。 */
    scanf("%d", &n);

    /* 接下来输入n个整数到a数组中去。 */
    for (i = 0; i < n; i++)
    {
        scanf("%d", &a[i]);
    }

    /* 不断缩小寻找最大值的范围。每次范围为: [left ~ n - 1] */
    for (left = 0; left <= n - 2; left++)
    {
        /* 假设 a[left] 为数组中最大的。 */
        m = left;

        /* 依次拿数组中值和假设的最大值位置比较, 如果找到更大的, 则替换假设最大值位置。 */
        for (i = left + 1; i < n; i++)
        {
            if (a[i] > a[m])
            {
                m = i;
            }
        }

        /* 交换 a[left] 和 a[m] */
        t = a[left];
        a[left] = a[m];
        a[m] = t;
    }

    /* 输出排序后的数组 */
    for (i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
    printf("\n");

    return 0;
}

```

7.2 二维数组

1. 二维数组的定义

类型 数组名[常量表达式1][常量表达式2];

例如: `int a[3][4];` 表示一个 3 行 4 列的数组。

2. 二维数组的取值和赋值

`a[0][1] = 5;` 元素赋值

`c = a[0][1];` 元素取值

3. 二维数组的初始化

```
int a[2][3] = {
    {1,2,3},
    {4,5,6}
};
```

7.3 多维数组

7.4 字符数组和字符串

1. 字符数组的定义

`char c[100];`

2. 字符数组初始化

`char c[10] = {'H', 'e', 'l', 'l', 'o'};`

`char c[10] = "Hello";`

3. 字符串和字符数组

C 语言中没有字符串类型, 字符串是通过字符数组来存储的。那么在存储中, 通过增加一个 `\0` 的标记位表示字符串在字符数组中存储的结束标记。

4. 字符数组元素的取值和赋值

```
char c[10] = "Hello";
char v;
v = c[0]; /* 取出 c 中下标为 0 的值, 并赋值给 v, 这时 v 为 'H' */
c[0] = 'A'; /* 向 c 中下标为 0 的位置赋值 'A', 这时字符串变为 "Aello" */
```

5. `scanf` 输入字符串

```
char str[100];
scanf("%s", str);
```

`scanf` 在输入时是会被空格截断的, 也就是说如上代码输入时是 `hello world`, 因为中间有一个空格, `scanf` 只会读取第一段 `hello`

6. printf 输出字符串

```
printf("{s}\n", str);
```

7. 字符串处理函数

- gets() 按行读取字符串
- puts() 输出字符串
- strlen() 计算字符串长度
- strcmp() 字符串比较函数。根据两个参数的小于、等于、大于关系返回 负数、0、正数
- strcpy() 字符串拷贝函数，将参数中第二个拷贝给第一个。
- strcat() 字符串拼接

第8章 函数

1. 函数的定义

```
int max(int a, int b)
{
    if (a > b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
```

2. 递归函数

```
int sum(int n)
{
    if (0 == n)
    {
        return 0;
    }

    return sum(n - 1) + n;
}
```

第 9 章

1. 指针的声明

```
int *p;
```

2. 指针的使用

```
#include <stdio.h>

int main()
{
    int a;
    int *p;

    a = 1;
    p = &a;
    printf("&a = %p\n", &a);
    printf(" p = %p\n", p);
    printf(" a = %d\n", a);
    printf("*p = %d\n", *p);

    *p = 10;
    printf(" a = %d\n", a);
    printf("*p = %d\n", *p);

    return 0;
}
```

3. 指针应用之交换变量值

```
#include <stdio.h>
#include <stdlib.h>

void swap(int *pa, int *pb)
{
    int t;
    t = *pa;
    *pa = *pb;
    *pb = t;
}

void sort(int *pa, int *pb, int *pc)
{
    if (*pa > *pb) swap(pa, pb);
    if (*pa > *pc) swap(pa, pc);
    if (*pb > *pc) swap(pb, pc);
}

int main()
{
}
```

```
int a, b, c;  
scanf("%d%d%d", &a, &b, &c);  
sort(&a, &b, &c);  
printf("%d %d %d\n", a, b, c);  
return 0;  
}
```

4. ...