广东白云学院 大数据协会

3：函数，条件分支

目标

- 学习函数式编程思维
- 学习条件分支语句

函数

- 一个有9000行代码的大型程序，如何让其他人在最快速度内搞明白这个程序的功能?

函数

- 一个有9000行代码的大型程序，如何让其他人在最快速度内搞明白这个程序的功能?

> 使用函数。将不同的方法封装进不同的函数里，在主函数中调用这些函数。

函数结构

```
返回值类型 函数名(函数参数1, 函数参数2, ...)
{
    函数体
    return 返回值
}
```

函数

```c
#include <stdio.h>

int findAverage(int num1, int num2); // 声明函数

int main(void) // 主函数，C程序从main()开始执行
{
    int avg;
    avg = findAverage(30, 20); // 调用函数
    printf("The average of 30 and 20: %d\n", avg);
    // 也是调用函数，printf()在<stdio.h>中声明
    printf("The average of 4 and 70: %d\n",
      findAverage(4, 70));
    return 0;
}
```

```c
int scanf(const char *format, ...) {
    va_list ap;
    int retval;

    va_start(ap, format);
    retval = _doscan(stdin, format, ap);
    va_end(ap);
    return retval;
}
```

```c
int printf(const char *fmt, ...) {
    int i;
    char buf[256];

    va_list arg = (va_list)((char*)(&fmt) + 4);
    i = vsprintf(buf, fmt, arg);
    write(buf, i);

    return i;
}
```

函数

```c
#include <stdio.h>

int findAverage(int num1, int num2); // 声明函数

int main(void) // 主函数，C程序从main()开始执行
{
    int avg;
    avg = findAverage(30, 20); // 调用函数
    printf("The average of 30 and 20: %d\n", avg);
    // 也是调用函数，printf()在<stdio.h>中声明
    printf("The average of 4 and 70: %d\n",
      findAverage(4, 70));
    return 0;
}
```

函数

```c
#include <stdio.h>

int findAverage(int num1, int num2); // 声明函数

int main(void) { // 主函数，C程序从main()开始执行
    int avg;
    avg = findAverage(30, 20); // 调用函数
    ...
}

int findAverage(int num1, int num2) { // 函数
    // 函数体
    int average = (num1 + num2) / 2;
    return average;
}
```

函数式编程

- 程序功能分割成许多个函数，减少程序复杂度，方便分工
- 使整个程序结构清晰
- 不写重复的代码

函数

- 每个函数只做1件事
- 使用函数的人不需要关心函数是怎么写的
- 模块化，便于测试与发现BUG

函数：使整个程序结构清晰

```
/* 编写一个程序，计算每个同学每门课程的绩点 */
int main(void)
{
    studentInfo = readData(); // 读取学生信息
    result = calculateGradePoint(studentInfo); // 计算课程绩点
    display(result); // 展示结果
    return 0;
}

readDate()、calculateGradePoint()、display()的代码...
```

函数：不写重复代码

```c
void meeting(char[] name) {
    printf("Hello, %s!\n", name);
}

int main(void) {
    meeting("Association of Big Data");
    meeting("Zhang San");
    meeting("C Programming Language");
    return 0;
}
```

```
Hello, Association of Big Data!
Hello, Zhang San!
Hello, C Programming Language!
```

Coding Time

Assn01 编写一个程序，实现一个函数：
输入直角三角形两条直角边，计算三角形斜边的长度

> 提示：开根使用sqrt()

Assn01 计算直角三角形斜边

```c
#include <stdio.h>
#include <math.h>
double hyp(double baseSide, double height);

int main(void) {
    printf("直角边为3和4的直角三角形斜边：%lf",
      hyp(3.0, 4.0));
    printf("直角边为6.7和7.4的直角三角形斜边：%lf",
      hyp(6.7, 7.4));
    return 0;
}

double hyp(double baseSide, double height) {
    return sqrt(baseSide*baseSide + height*height);
}
```

条件分支

```c
#include <stdio.h>
void printBiggerOne(int num1, int num2) {
    if (num1 > num2) {
        printf("The Bigger One: %d", num1);
    }
    else if (num1 == num2) {
        printf("%d equals to %d.\n", num1, num2);
    }
    else {
        printf("The Bigger One: %d", num2);
    }
}

int main(void) {
    int num1, num2;
    num1 = 8; num2 = 3;
    printBiggerOne(num1, num2);
    return 0;
}
```

Coding Time

Assn02 增加功能：打印比较小的数

```
#include <stdio.h>
void printBiggerOne(int num1, int num2) {
    if (num1 > num2) {
        printf("The Bigger One: %d", num1);
    }
    else if (num1 == num2) {
        printf("%d equals to %d.\n", num1, num2);
    }
    else {
        printf("The Bigger One: %d", num2);
    }
}
int main(void) {
    int num1, num2;
    num1 = 8; num2 = 3;
    printBiggerOne(num1, num2);
    return 0;
}
```

广东白云学院 大数据协会

感谢聆听！