

函数

2018年6月10日 20:25

函数=方法

定义在类中的具有特定功能的小程序

```
class FunctionDemo
```

```
{
```

```
    public static void main(String[] args)
```

```
    {
```

```
        int c = add(3,4);
```

```
        System.out.println("c="+c);
```

```
        myprint();
```

```
    }
```

```
    /*
```

定义函数的格式：

修饰符 返回值类型 函数名 (参数类型：参数1, 参数类型：参数2)

```
    {
```

```
        执行语句;
```

```
        return 返回值; //return 可以用来结束函数
```

```
    }
```

特殊情况：功能没有具体返回值

这时return的后面直接用分号结束，关键字是void

注意：如果返回值类型为void，return语句可以省略不写

***不要在函数中定义函数

```
    */
```

```
    public static void myprint()
```

```
    {
```

```
        System.out.println("hello java");
```

```
        int c =9;
```

```
        //return ;
```

```
    }
```

```
    static int add(int a,int b)
```

```
    {
```

```
        return a+b;
```

```
    }
```

```
    /*
```

```
public static void add(int a, int b)
{
    System.out.println(a+b);
}
该方法不好，只需运算即可，没有必要输出
*/
}
```

练习1

2018年6月13日 21:05

class FunctionDemo2

{

public static void main(String[] args)

{

draw(4,6);

draw(7,9);

}

/*

需求1：定义一个功能，完成两个数和的获取

思路：既然定义功能，就可以用函数来实现

通过两个明确来完成

明确一：这个功能的结果是什么？ int

明确二：这个功能实现过程中是否需要未知内容参与运算？ 加数和被加数（参数列表）

注意：返回值类型与参数类型没有关系

*/

public static int add(int a,int b)

{

return(a+b);

}

/*

需求2：画一个矩形在控制台

*/

public static void draw(int row, int col)

{

for(int x=1;x<=row;x++)

{

for(int y = 1;y<col;y++)

{

System.out.print("*");

}

System.out.println();

}

}

/*

需求3：比较两个数是否相等

*/

```

public static boolean equals(int a, int b)
{
    /*
    if(a==b)
        return true;
    return false;
    */
    //return(a==b)?true:false;
    return a==b;
}
/*
需求4: 定义一个功能, 获取两数中较大数
*/
public static int getMax(int a,int b)
{
    return a>b?a:b;
}
}

```

练习2

2018年6月13日 21:42

```
class FunctionDemo3
{
    public static void main(String[] args)
    {
        print99();

    }
    /*
    打印九九乘法表
    */
    public static void print99()
    {
        for(int x = 1;x<=9;x++)
        {
            for(int y=1;y<=x;y++)
            {
                System.out.print(y+"*"+x+"="+x*y+"\t");
            }
            System.out.println();
        }
    }
    /*
    打印成绩等级
    */
    public static char getLevel(int num)
    {
        char level;
        if(num>=90 && num<=100)
            level = 'A';
        else if(num>=80 && num<=89)
            level = 'B';
        else if(num>=70 && num<=79)
            level = 'C';
        else if(num>=60 && num<=69)
            level = 'D';
        else
```

```
        level = 'E';  
    return level;  
    }  
}
```

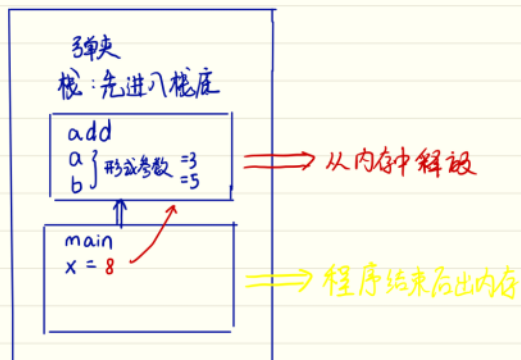
底层运算原理

2018年6月13日 22:30

底层运算原理

```
class FunctionDemo2
{
    public static int add(int a, int b)
    {
        return a+b;
    }
    public static void main(String[] args)
    {
        int x = add(3, 5);
        System.out.println("x="+x);
    }
}
```

编译 不找 main 主函数
↓ 运行 找 main (否则 error)
java FunctionDemo2



函数的重载

2018年6月13日 22:54

Overload 重新载入

在同一类中，允许存在一个以上的同名函数，只要他们的参数个数或参数类型不同即可

```
class FunctionDemo4
{
    public static void main(String[] args)
    {
        double x=add(1.1,2.2);
        int y=add(1,2);
        int z=add(1,2,3);
        System.out.println(x);
    }
    /*
    函数的重载
    函数重载与返回值类型无关
    java是严谨性语言
    */
    //加法运算：两个整数的和
    public static int add(int a,int b)
    {
        return a+b;
    }
    //加法运算：两个小数的和
    public static double add(double a,double b)
    {
        return a+b;
    }
    //加法运算：三个整数的和
    public static int add(int a,int b,int c)
    {
        return a+b+c;
    }

    public static void print99(int num)
    {
        for(int x=1; x<=num; x++)
```



```

        {
            for(int y=1; y<=x; y++)
            {
                System.out.print(y+"*"+x+"="+y*x+"\t");
            }
            System.out.println();
        }
    }

    //打印99乘法表
    public static void print99()
    {
        print99(9);
    }
}

```