

Python程序设计

第六讲 函数与模块化 概述



张 华
WHU

程序模块化与函数

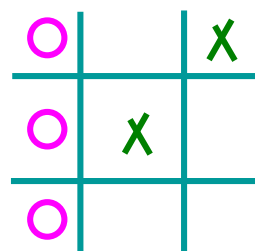
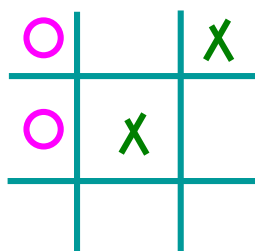
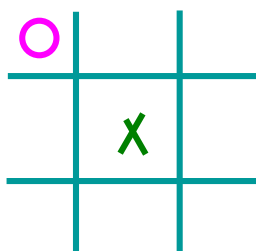
- 程序模块化
- 函数的概念
- Python函数的类型

引例

问题：“井”字棋游戏程序

游戏规则：

- 玩家和计算机轮流下棋子。
- 每次一方只能在某一个空格处下一颗棋子。
- 胜负判断：
 - 若棋盘的某一行，或某一列，抑或某一对角线上的三个格子被某一方的棋子占据，则该方胜利；
 - 否则，为平局。

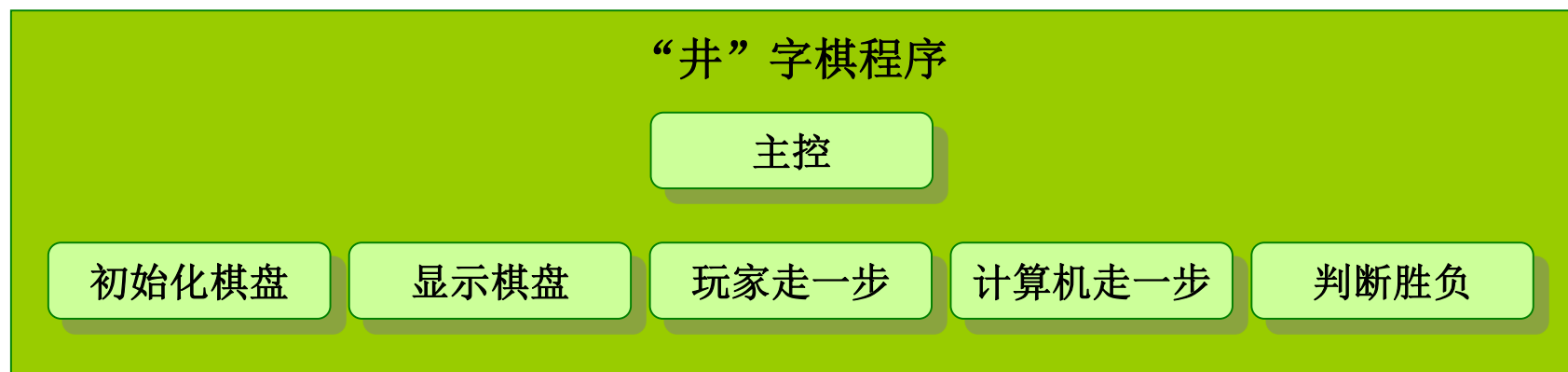


程序模块化

结构化软件的开发方法

- ✿ 分解：采用模块化结构，将一个复杂的任务分解为若干相对简单并彼此较独立的模块，还可以将这些模块再细分为若干更小的子模块，以利于“**分而治之**”，实现**自顶向下**的结构化设计；模块可重复使用。
- ✿ 抽象：允许使用功能代码，不需要考虑其内部的实现。

函数是实现结构化程序设计必不可少的工具。



函数

基本思想

- ✿ 将可能需要反复执行的代码封装为函数，并在需要该功能的地方进行调用。
- ✿ 不仅可以实现代码复用，更重要的是可以保证代码的一致性，只需要修改该函数代码则所有调用均受到影响。

设计函数时，应注意提高模块的内聚性，同时降低模块之间的隐式耦合。

通过函数实现程序模块化

通过函数实现程序模块化的更多好处：

✿ 减少程序的复杂度。

- 简化程序的结构，提高程序的可阅读性。

✿ 实现代码的复用。

- 一次定义多次调用，实现代码的可重用性。

✿ 提高代码的质量。

- 实现分割后子任务的代码相对简单，易于开发、调试、修改和维护。

✿ 协作开发。

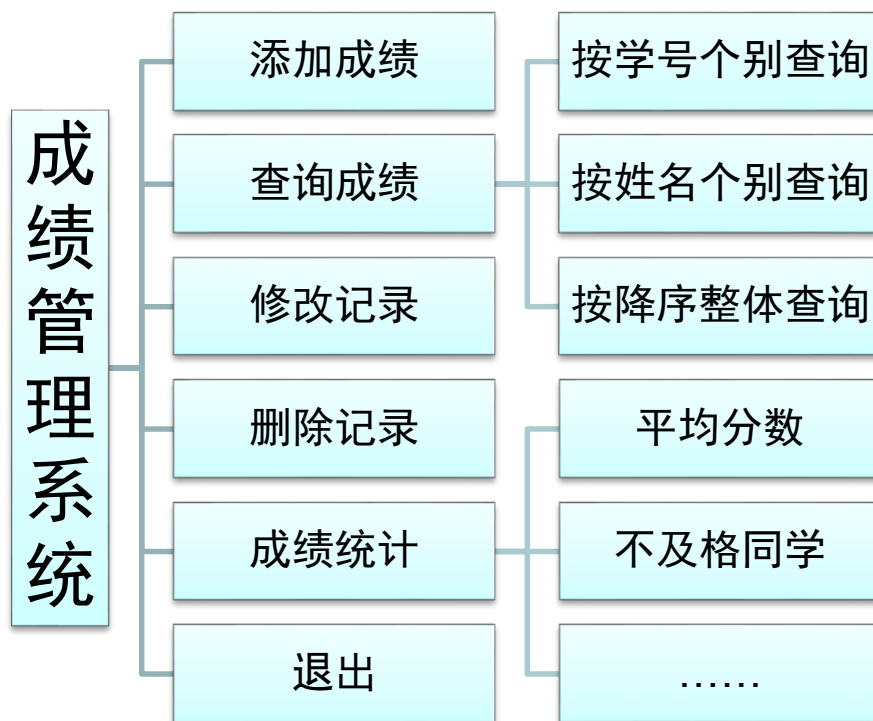
- 大型项目分割成不同的子任务后，团队多人可以分工合作，同时进行协作开发。

✿ 实现特殊功能。

- 递归函数可以实现许多复杂的算法。

通过函数实现程序模块化

■ 模块化后的程序结构示例



Python函数的类型

■ 内置函数

- ✿ Python语言内置了若干常用的函数，例如`abs()`、`len()`等等，在程序中可以直接使用。

■ 标准库函数

- ✿ Python语言安装程序同时会安装若干标准库，例如`math`、`random`等等。
- ✿ 通过`import`语句，可以导入标准库，然后使用其中定义的函数。

Python函数的类型

第三方库函数

- ✿ Python社区提供了许多其他高质量的库，如Python图像库等等。
- ✿ 下载安装这些库后，通过import语句，可以导入库，然后使用其中定义的函数。

用户自定义函数

- ✿ 根据解决问题的需要，程序开发人员自己设计和定义的函数。