

Python程序设计

案例：猜数游戏程序



张 华

WHU

“猜数” 游戏程序

- 问题
- 需求分析
- 设计
- 实现
- 改进

结构化程序设计案例

问题：编写“猜数”游戏程序

- ✿ 程序产生一个1~100之间的随机数。
- ✿ 用户输入一个数。
- ✿ 程序判断用户是否猜对了，否则给出恰当的提示，让用户继续猜，直到猜对为止。



结构化程序设计案例

需求分析+设计

✿ 需要什么数据？如何表示？——设计数据结构

- 随机数
- 用户猜的数
- 其他

✿ 如何处理这些数据？——设计算法

- 产生随机数
- 提示用户输入所猜的数
- 判断是否猜对了，正确则提示并结束，否则转到第2步

结构化程序设计案例

实现

基础版代码

```
import random

secretNumber = random.randint(1,100)
guessedNumber = -1;

print('有一个1~100内的数，请你猜一猜。')
while guessedNumber != secretNumber:
    guessedNumber = int(input('输入你想到的数: '))

    if guessedNumber != secretNumber:
        print('不对哟，再试试。')

print('祝贺你猜对了!')
```

程序设计

改进“猜数”游戏程序

✿ 提高可玩性

- 如果用户猜的不对，给用户更多的提示，例如“你猜的数太大了”。

✿ 提高趣味性

- 统计用户猜的次数，并给出评价。

✿ 增强功能

- 一次游戏结束后，询问用户是否再玩一次。

程序设计

实现

提高版代码

```
import random

secretNumber = random.randint(1,100)
guessedNumber = -1;

print('有一个1~100内的数，请你猜一猜。')
while guessedNumber != secretNumber:
    guessedNumber = int(input('输入你想到的数: '))

    if guessedNumber > secretNumber:
        print('大了，再试试。')
    elif guessedNumber < secretNumber:
        print('小了，再试试。')

print('祝贺你猜对了！')
```

程序设计

实现

继续改进

```
import random

secretNumber = random.randint(1,100)
guessedNumber = -1;

print('有一个1~100内的数，请你猜一猜。')
while guessedNumber != secretNumber:
    guessedNumber = int(input('输入你想到的数：'))

    deviation = abs(guessedNumber - secretNumber)
    if deviation >=10:
        print('太', end='')
    elif deviation >=5:
        print('有点', end='')
    elif deviation != 0:
        print('稍微', end='')

    if guessedNumber > secretNumber:
        print('大了，再试试。')
    ...
```


程序设计

追求更炫的效果

不断改进.....

➤ 直到实现比较自然的交互：语音+动画

开始猜数



不对噢，太大啦



好棒，猜对啦

37

19



程序设计

■ 遐想.....

✿ 计算机自动猜数

- 自动产生数，或者由人输入数，然后计算机模拟人去猜。
- 需要设计猜数算法，以最少的次数猜对。

✿ 用于行为研究

- 对用户猜数的行为特点进行分析。
 - 例如，猜第一个数的偏好
 - 例如，调整的幅度，有没有规律。