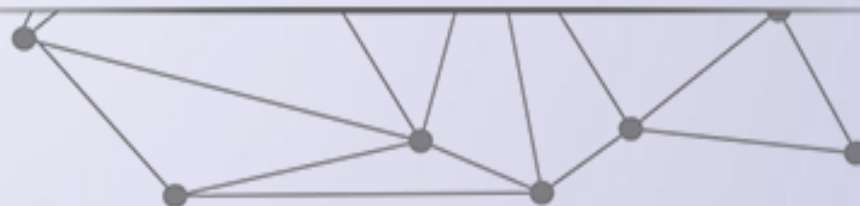




# 面向过程的程序设计



黄天羽

北京理工大学





# 面向过程的程序设计

- 以程序执行过程为设计流程的思想
- 程序设计中最自然的一种设计方法
- 结构化编程



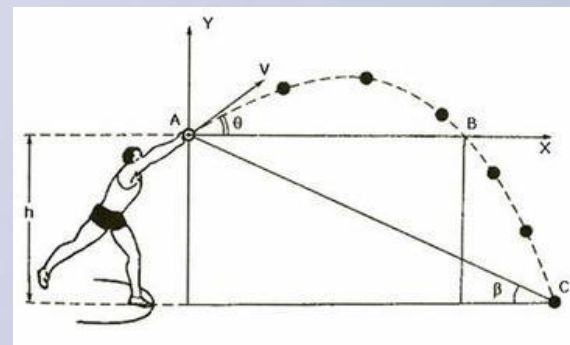
## 举例

- 新闻报道：2015年8月22日，世界锦标赛揭开战幕，德国选手施瓦尼茨以20米37夺冠，中国选手巩立娇以20米30获得亚军。
- 评论员评论分析：
  - 巩立娇是技术优秀型选手
  - 施瓦尼茨是力量型选手



# 铅球飞行计算问题

- 在给定不同的投掷角度和初始速度下，求解计算铅球的飞行距离
- IPO描述为：
  - 输入：铅球发射角度、初始速度(m/s)、初始高度(m)
  - 处理：模拟铅球飞行，时刻更新铅球在飞行中的位置
  - 输出：铅球飞行距离(m)





# 简化问题

- 忽略空气阻力
- 重力加速度  $9.8 \text{ m/s}^2$
- 铅球飞行过程
  - 铅球高度
  - 飞行距离



- 时刻更新铅球在飞行中的位置。
  - 假设起始位置是点  $(0,0)$
  - 垂直方向上运动距离 ( $y$ 轴)
  - 水平方向上移动距离 ( $x$ 轴)



# 设计参数：

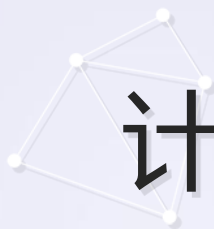
- 仿真参数：投掷角度angle、初始速度velocity、初始高度height、飞行距离interval
- 位置参数：x轴坐标xpos，y轴坐标ypos
- 速度分量：x轴方向上速度xvel，y轴方向上速度yvel



# 根据提示输入仿真参数

```
angle = eval(input("Enter the launch angle (in degrees):"))  
vel = eval(input("Enter the initial velocity (in meters/sec):"))  
h0 = eval(input("Enter the initial height (in meters):"))  
time = eval(input("Enter the time interval: "))
```





# 计算初始速度：

- x轴的速度

$$xvel = velocity * \cos(\theta)$$

- y轴的速度

$$yvel = velocity * \sin(\theta)$$



```
from math import pi, sin, cos, radians
```

```
xpos = 0  
ypos = h0
```

```
theta = radians(angle)  
xvel = vel * cos(theta)  
yvel = vel * sin(theta)
```



## 程序主循环：

```
while ypos >= 0:  
    xpos = xpos + time * xvel  
    yvell = yvel - time * 9.8  
    ypos = ypos + time * (yvel + yvell)/2.0  
    yvel = yvell
```



# 完整代码：

```
from math import pi, sin, cos, radians

def main():
    angle = eval(input("Enter the launch angle (in degrees):"))
    vel = eval(input("Enter the initial velocity (in meters/sec):"))
    h0 = eval(input("Enter the initial height (in meters):"))
    time = eval(input("Enter the time interval: "))

    xpos = 0
    ypos = h0

    theta = radians(angle)
    xvel = vel * cos(theta)
    yvel = vel * sin(theta)

    while ypos >= 0:
        xpos = xpos + time * xvel
        yvell = yvel - time * 9.8
        ypos = ypos + time * (yvel + yvell)/2.0
        yvel = yvell
    print("\nDistance traveled:{0:0.1f}meters.".format(xpos))
```



# 程序模块化：

```
def main():  
    angle, vel, h0, time = getInputs()  
    xpos, ypos = 0, h0  
    xvel, yvel = getXYComponents(vel, angle)  
    while ypos >= 0:  
        xpos, ypos, yvel = updatePosition(time, xpos, ypos, xvel, yvel)  
    print("\nDistance traveled: {0:0.1f} meters.".format(xpos))
```





# 面向过程程序设计基本步骤：

- 分析程序从输入到输出的各步骤
- 按照执行过程从前到后编写程序
- 将高耦合部分封装成模块或函数
- 输入参数，按照程序执行过程调试



# 总结面向过程程序设计特点

- 通过分步骤、模块化
  - 将一个大问题分解成小问题
  - 将一个全局过程分解为一系列局部过程
- 面向过程
  - 最为自然、也是最贴近程序执行过程的程序设计思想
  - 在面向对象的程序设计中也会使用面向过程的设计方法

