程序设计（Python）

分支结构

主讲：数据与目标工程学院 胡瑞娟 副教授

知识回顾

【字符串案例】在军事通信中，信息的格式是生命线。一个格式错误的坐标、一条含义模糊的指令，可能导致任务失败。今天，我们的任务就是扮演一名情报分析员，处理混乱的原始战场数据，并将其格式化为清晰、标准的战术指令(情报摘要）。具体来说：你截获了多段来自不同单位的战场通信文本。这些文本格式混乱、包含冗余代码和干扰字符。如："ALPHA7|n39-52.783,e116-25.467|T0830|CONFIRMED"
你的任务是：

- 清洗和标准化文本，去除干扰符号
- 提取关键信息（单位代号、坐标、时间、**状态**）
- 将坐标格式标准化
- 生成清晰的情报摘要

字符串

1.字符串**概念** - 认识什么是字符串
2.字符串**长度** - 了解信息规模
3.字符串**分割** - 拆解复杂信息
4.字符串**查找/切片** - 定位关键内容
5.字符串**替换** - 清洗和标准化格式
6.字符串**拼接/输出** - 生成最终报告

知识回顾

输入 → 处理 → 输出

字符串定义

字符串长度
↓
字符串分割
↓
字符串查找/切片
↓
字符串替换
↓
字符串拼接/输出 - 生成最终报告

```
========== 生成情报报告 ==========
【情报报告】 - 军事通信
单位：ALPHA-7
坐标：N3952.783,E11625.467
时间：0830
状态：CONFIRMED
==============================

【坐标详细分析】
纬度：N 3952.783
经度：E11625.467
格式化：N3952.783,E11625.467
```

顺序结构

案例引入

【案例】基于上述案例（军事通信信息），我们需要**根据不同的通信内容做出不同的处理**。比如：根据状态，

（1）如果是紧急通信，要立即上报；

（2）如果是紧急通信，要立即上报；如果是常规通信，按正常流程处理。

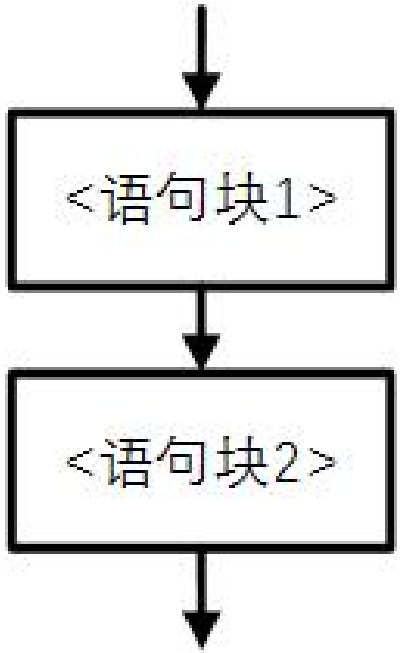
（3）如果是紧急通信，要立即上报；如果是常规通信，10分钟内处理；如果是待处理通信，30分钟内处理；其他情况，标记为待核实，暂不处理。

分支结构
（选择）

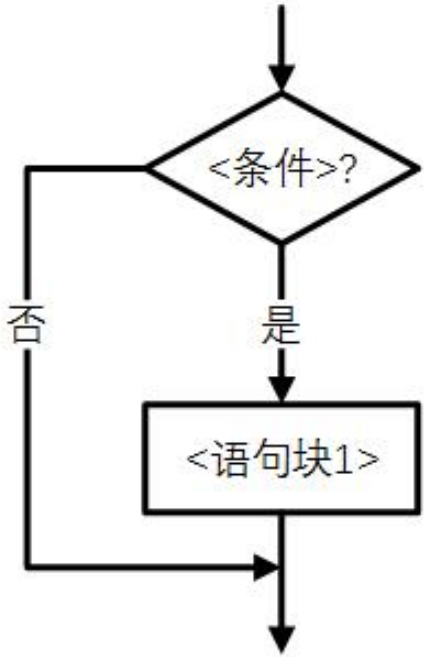
01 程序基本结构

- 程序的基本结构是指构成程序的不同类型的控制流。
- 共有三种：
  - 顺序结构
  - 分支结构
  - 循环结构

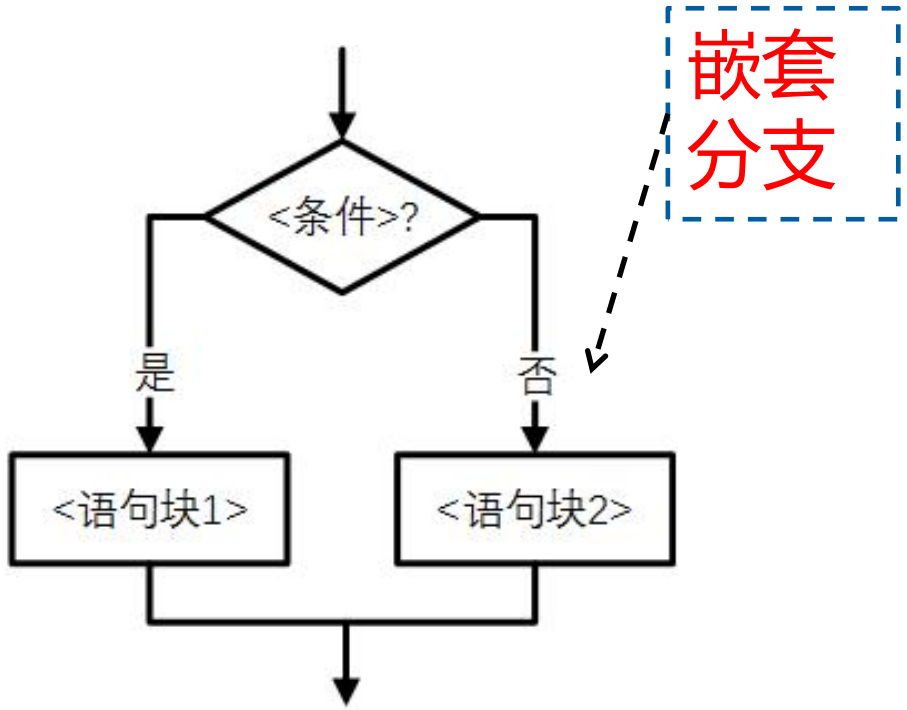
- 顺序结构是程序按照线性顺序依次执行的一种运行方式，如图所示，其中语句块1和语句块2表示一个或一组顺序执行的语句。



- 分支结构是程序根据条件判断结果而选择不同执行路径的一种运行方式。由二分支结构会组合形成多分支结构。**（你能举出生活中的例子吗？）**



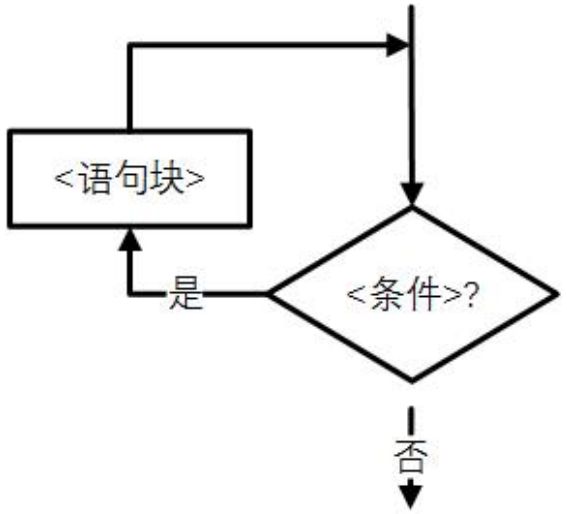
单分支结构



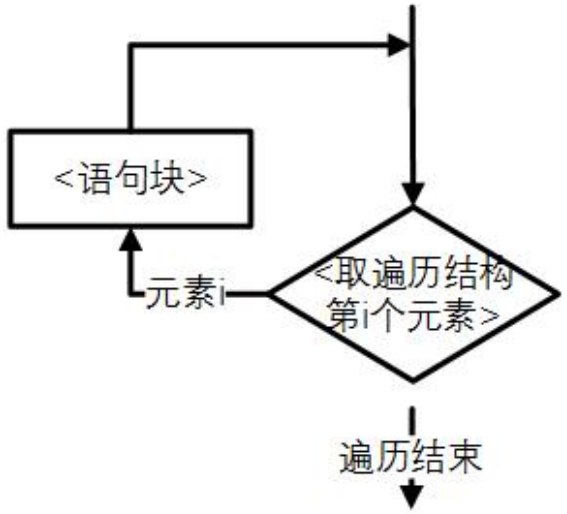
二分支结构

- 循环结构是程序根据条件判断结果后反复执行的一种运行方式，根据循环体触发条件不同，循环结构包括条件循环和遍历循环结构。

新训走齐步



条件循环



遍历循环

流水作业

- 程序（算法）的描述方式主要有：自然语言、流程图、伪代码、代码。
  - 自然语言描述方式指使用人类语言直接描述程序，IPO方法是这类的一种。优点是灵活自然，缺点是比较繁琐，容易出现二义性，一个描述可以产生多种不同的程序代码。
  - 流程图描述是程序最直观易懂的表达方式，主要适用于较短算法。优点是直观、清晰且逻辑确定，缺点是流程图绘制比较繁琐，当程序较大时流程图会很复杂，反而降低了表达的清晰性。
  - 伪代码描述是介于自然语言与编程语言之间的一种算法描述语言。使用伪代码不用拘泥于具体编程语言，对整个算法运行过程的描述最接近自然语言。与直接的自然语言描述不同，伪代码在保持程序结构的情况下描述算法。

【示例】根据学员的分数判定成绩是否合格。

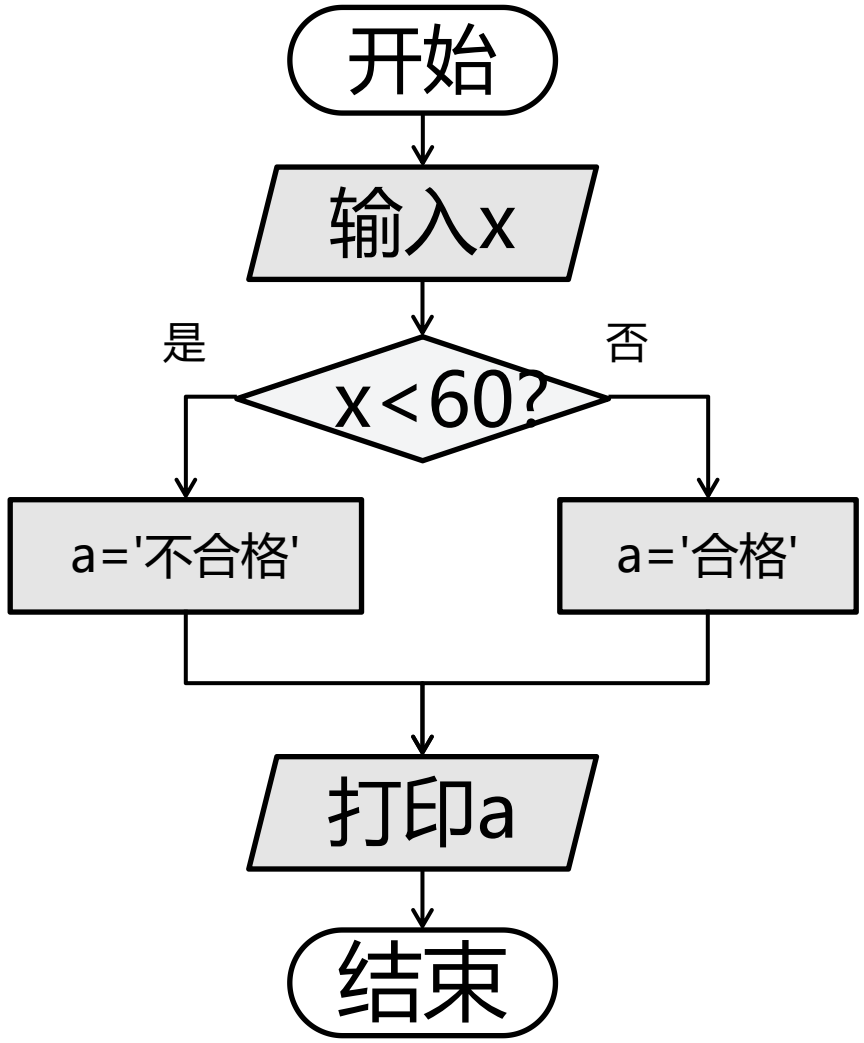
```
x=eval(input('x='))
  分情况讨论
  根据分数是否小于60
  判定合格与否
print('学员成绩', a)
```

请用户输入一个分数

根据分数是否小于60判定合格与否

打印这个判定结果

自然语言



流程图

输入 x

若 x<60: a设为不合格
否则: a设为合格

输出 a

伪代码

流程图（程序流程图）

用一系列图形、流程线和文字说明描述程序的基本操作和控制流程，它是程序分析和过程描述的最基本方式。

流向线以带箭头直线或曲线形式指示程序的执行路径；
连接点将多个流程图连接到一起，常用于将一个较大流程图分割为若干部分。

【案例分析】

问题1：根据状态，如果是紧急通信，要立即上报！

"ALPHA7|N39-52.783,E116-25.467|T0830|**URGENT**"

```
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|URGENT
通信状态: URGENT
!!!发现紧急通信!
立即启动优先处理通道
通知值班指挥官
继续监控其他通信...
```

```
# 接收通信信息
message = "ALPHA7|N39-52.783,E116-25.467|T0830|URGENT"
print("收到通信:", message)
# 解析通信内容（使用上节课的字符串知识）
parts = message.split('|')
status = parts[3]  # 状态信息在第四个位置
print("通信状态:", status)
# 如果是紧急状态，怎么处理？？
```

02

Python 分支语句——if语句

【案例分析】

解决问题1：根据状态，如果是紧急通信，要立即上报！

"ALPHA7|N39-52.783,E116-25.467|T0830|**URGENT**"

```
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|URGENT
通信状态: URGENT
!!!发现紧急通信!
立即启动优先处理通道
通知值班指挥官
继续监控其他通信...
```


```
# 接收通信信息
message = "ALPHA7|N39-52.783,E116-25.467|T0830|URGENT"
print("收到通信:", message)
# 解析通信内容（使用上节课的字符串知识）
parts = message.split('|')
status = parts[3]  # 状态信息在第四个位置
print("通信状态:", status)
# 单分支结构：如果是紧急状态，特殊处理
if status == "URGENT":
    print("！！！发现紧急通信！")
    print("立即启动优先处理通道")
    print("通知值班指挥官")
print("继续监控其他通信...")
```

【案例分析】

问题2：如果是紧急通信，要立即上报；如果是常规通信，按正常流程处理。

"ALPHA7|N39-52.783,E116-25.467|T0830|**URGENT**"

```
=== 紧急通信识别系统 ===
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|URGENT
通信状态: URGENT
发现紧急通信,立即上报!
```

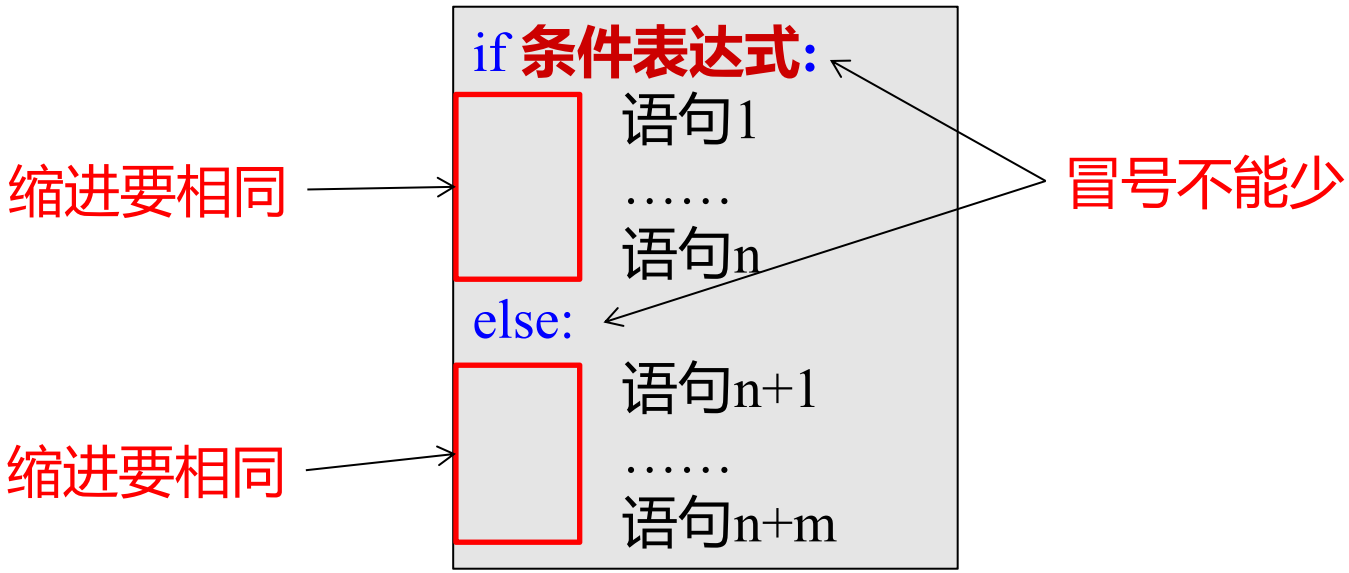
"ALPHA7|N39-52.783,E116-25.467|T0830|**CONFIRMED**"

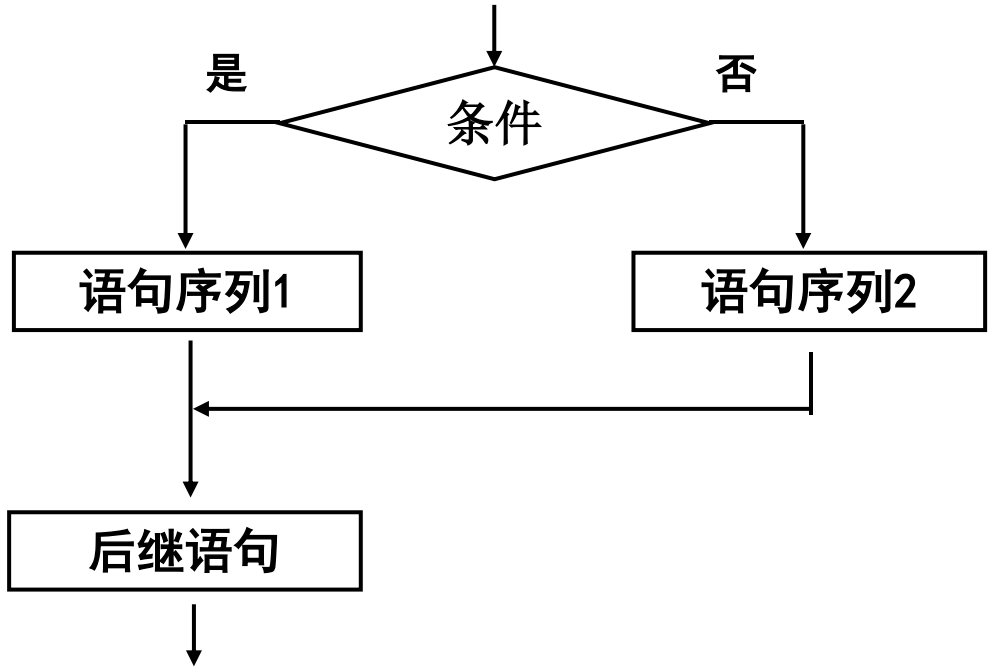
```
=== 紧急通信识别系统 ===
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED
通信状态: CONFIRMED
常规通信，按正常流程处理!
```

2.2二分支结构：if-else语句

- Python中if-else语句用来形成二分支结构，语法格式如下：

若**条件表达式**成立则执行**语句1~语句n**，否则执行**语句n+1~语句n+m**





【案例分析】

解决问题2：如果是紧急通信，要立即上报；如果是常规通信，按正常流程处理。

"ALPHA7|N39-52.783,E116-25.467|T0830|URGENT"

```
=== 紧急通信识别系统 ===
收到通信：ALPHA7|N39-52.783,E116-25.467|T0830|URGENT
通信状态：URGENT
发现紧急通信,立即上报!
```

```
=== 紧急通信识别系统 ===
收到通信：ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED
通信状态：CONFIRMED
常规通信，按正常流程处理!
```

```
if status == "URGENT":
    print("发现紧急通信，立即上报！")
else:
    print("常规通信，按正常流程处理！")
```

2.2二分支结构：if-else语句

- 二分支结构还有一种更简洁的表达方式，适合通过判断返回特定值，语法格式如下：

  <表达式1> if <条件> else <表达式2>

- 其中，<表达式1/2>一般是数字类型或字符串类型的一个值

```
if status == "URGENT":
    print("发现紧急通信，立即上报！")
else:
    print("常规通信，按正常流程处理！")
```

```
x=eval(input('x='))
if x<60:
    a='不合格'
else:
    a='合格'
print("学员成绩是",a)
```

```
a = "发现紧急通信，立即上报！" if status == "URGENT" else "常规通信，按正常流程处理！"
```

```
a = '不合格' if x<60 else '合格'
```

【案例分析】

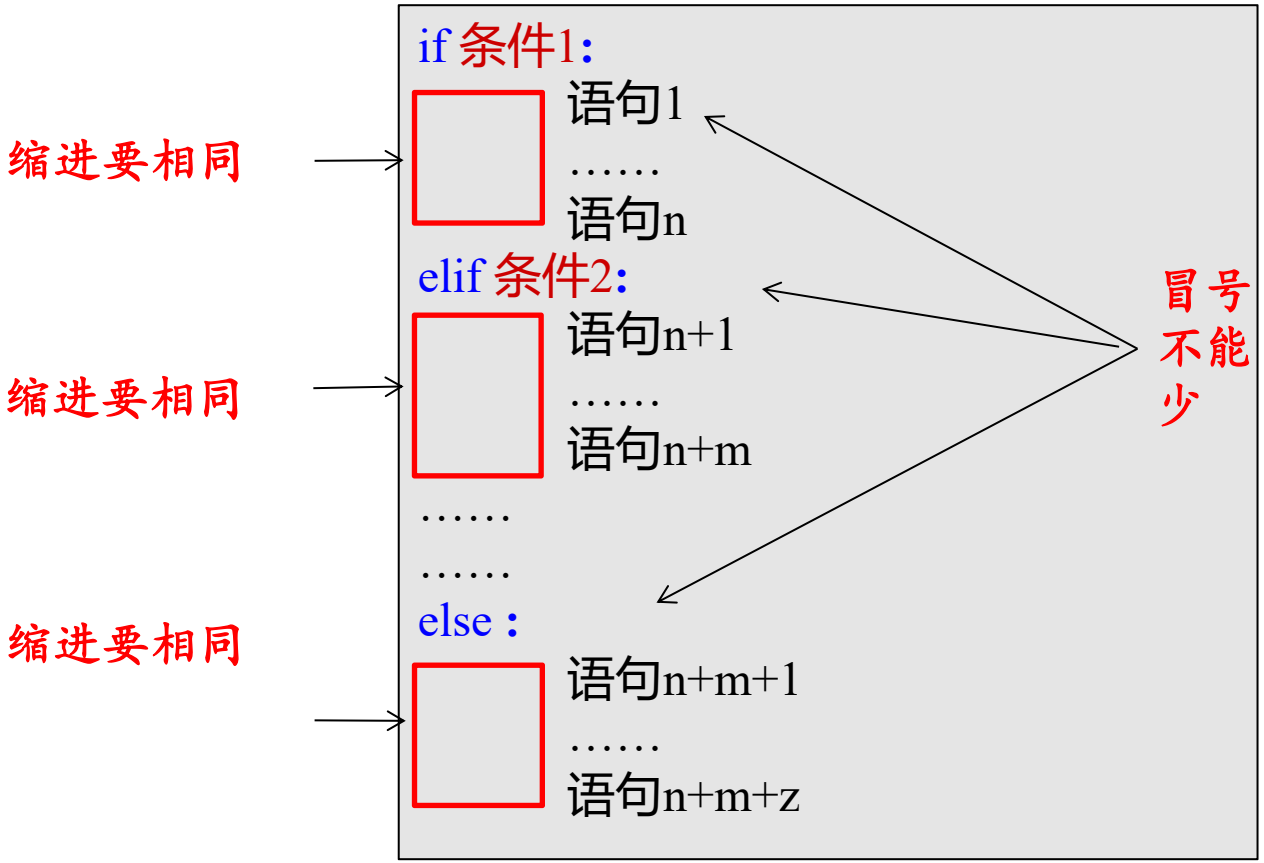
问题3：如果是紧急通信(URGENT)，要立即上报；如果是常规通信(CONFIRMED)，10分钟内处理；如果是待处理通信(PENDING)，30分钟内处理；其他情况，标记为待核实，暂不处理。

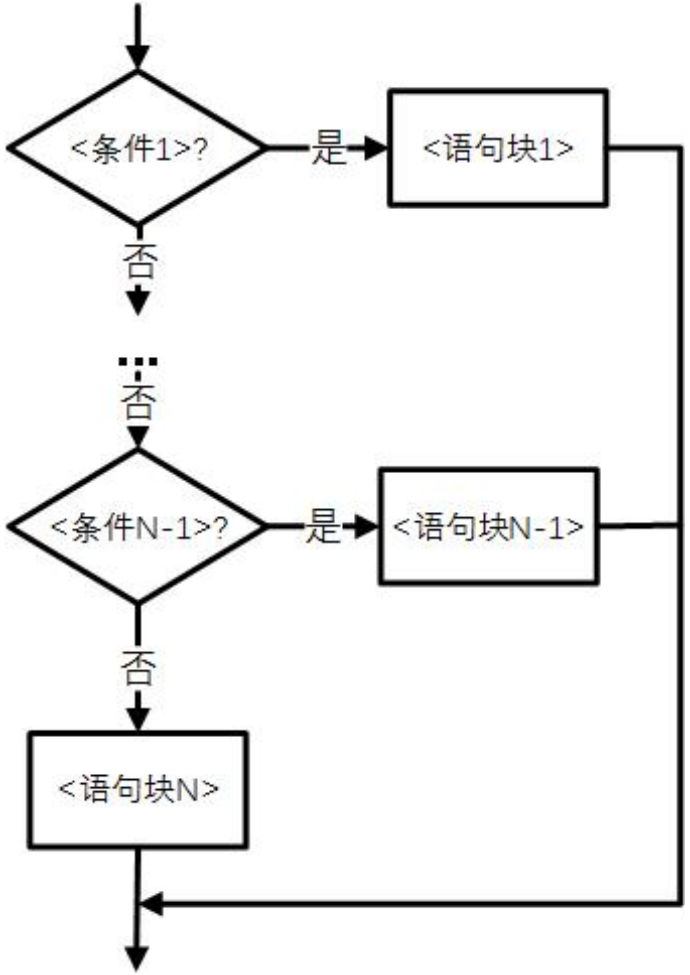
"ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED"

```
=== 紧急通信识别系统 ===
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED
通信状态: CONFIRMED
10分钟内处理
分类完成，等待下一条通信...
```

```
=== 紧急通信识别系统 ===
收到通信: ALPHA7|N39-52.783,E116-25.467|T0830|Other
通信状态: Other
标记为待核实,暂不处理
分类完成，等待下一条通信...
```

Python的if-elif-else描述多分支结构





【注意】多分支结构是二分支结构的扩展，通常**用于设置同一个判断条件的多条执行路径。**Python依次评估寻找第一个结果为True的条件，执行该条件下的语句块，同时跳过整个if-elif-else结构，执行后面的语句。如果没有任何条件成立，else下面的语句块被执行。else子句是可选的。

【案例分析】

解决问题3：如果是紧急通信(**URGENT**)，要立即上报；如果是常规通信(**CONFIRMED**)，10分钟内处理；如果是待处理通信(PENDING)，30分钟内处理；其他情况，标记为待核实，暂不处理。

"ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED"

```
if status == "URGENT":
    print("立即上报")
elif status == "CONFIRMED":
    print("10分钟内处理")
elif status == "PENDING":
    print("30分钟内处理")
else:
    print("标记为待核实,暂不处理")
print("分类完成，等待下一条通信...")
```

2.4条件件表达式

【思考】如果条件比较复杂，如何写出正确的**条件表达式**？

```
if status == "URGENT":
    print("发现紧急通信，立即上报！")
else:
    print("常规通信，按正常流程处理！")
```

【示例】报数——序号为1、3、5、7、9、10、12、14的学员出列，执行紧急任务，其余人员待命。

```
m=int(input('序号是: '))
if m==1 or m==3 or m==5 or m==7 or m==9
or m==10 or m==12 or m==14:
    print('执行紧急任务')
else:
    print('原地待命')
```

2.4条件件表达式

(1)比较运算符

- \> ：大于，如x>0
- < ：小于，如x<0
- >= ：大于等于，如x>=0
- <= ：小于等于，如x<=0
- == ：等于，如x==0
- != ：不等于，如x!=0

(2)逻辑运算符

- and：与（而且）
  - 例："**a>-1 and a<1**"表示" a>-1且a<1"
  - 注："**a>-1 and a<1**"也可写成"-1<a<1"
- or：或（或者）
  - 例："**a>1 or a<-1**"表示" a>1或a<-1 "
- not：非（不是）
  - 例："**not a>1**"表示" a不大于1"

(3)运算符优先级

```
print(2**2**3)
```

- 小括号具有最高优先级。

```
64? 256?
```

- 运算符优先级按类别排序：算术 > 比较 > 逻辑。

运算符优先级	算术运算符	**（右结合）
		*、/、%、//
		+、-
	比较运算符	<=、<、>、>=
		==、!=
	逻辑运算符	and or not（右结合）

从高到低

【分析程序运行结果】

```
a = 35
x = 2
if not x:
    a = 25
print(a)
```

一个四位数7249的个位、十位、百位、千位提取方法？比较得出各个位的最小值？

- 个位d：数字 % 10
- 十位c：(数字 // 10) % 10
- 百位b：(数字 // 100) % 10
- 千位a：数字 // 1000

【示例】报数——序号为1、3、5、7、9、10、12、14的学员出列，执行紧急任务，其余人员待命。

```
m=int(input('序号是: '))
if m==1 or 3 or 5 or 7 or 9 or 10 or 12
or 14:
    print('执行紧急任务')
else:
    print('原地待命')
```

```
m=int(input('序号是: '))
if (1<=m<=9 and m%2) or (10<=m<=15 and
m%2==0):
    print('执行紧急任务')
else:
    print('原地待命')
```

m在1~9之间且为奇数

设置

```
#I 输入
score=eval(input())
#P：多分支
if score>=60:
    print("D")
elif score>=70:
    print("C")
elif score>=80:
    print("B")
elif score>=90:
    print("A")
```

A A

B B

C C

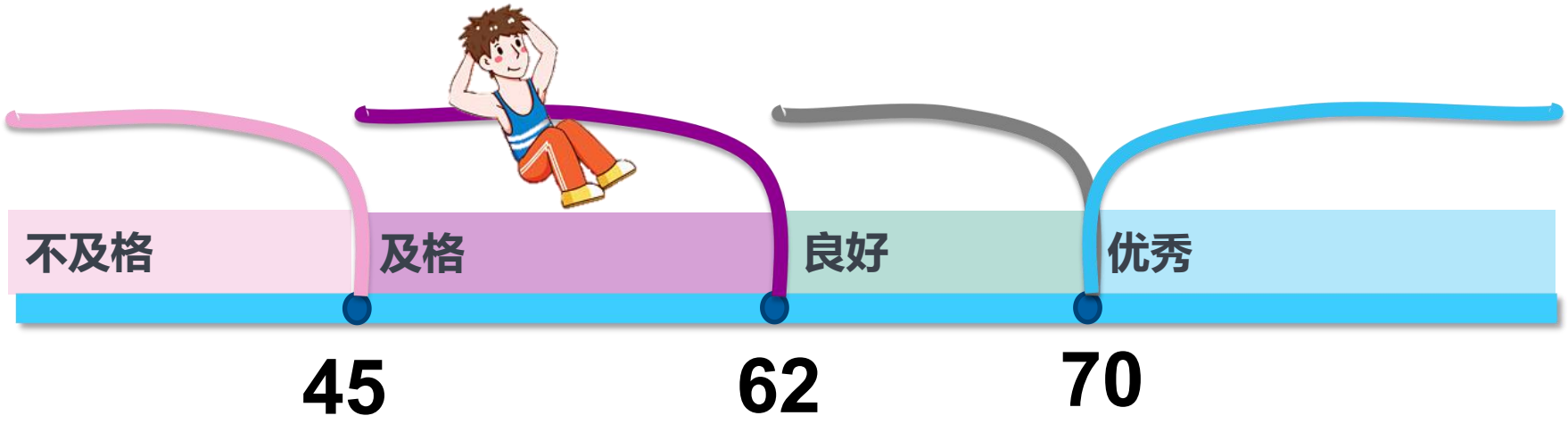
D D

从键盘输入81，则程序运行结果是（）

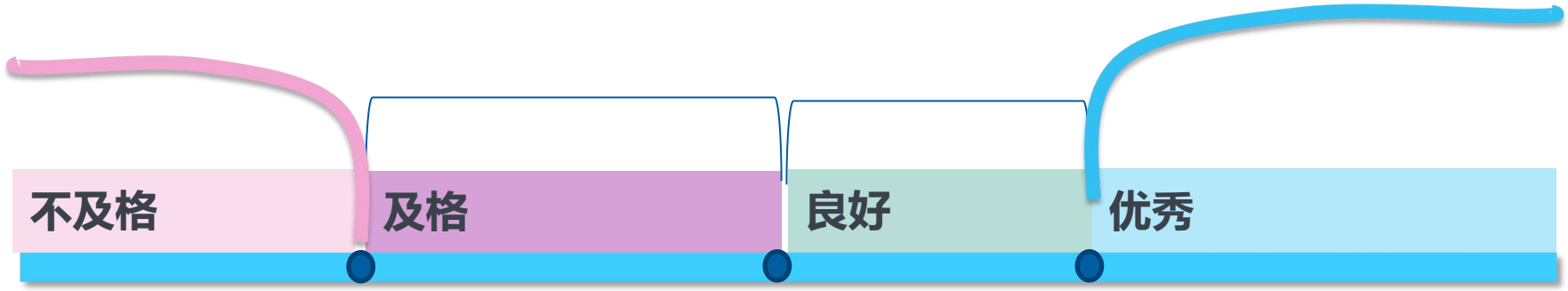
提交

【举一反三】体能考核：仰卧起坐。

求解重点：如何写出正确的**条件表达式**？



2.4条件件表达式



situpNum <45

situpNum >45 and situpNum <62

45 < situpNum <62

situpNum >62 and situpNum <70

situpNum >70

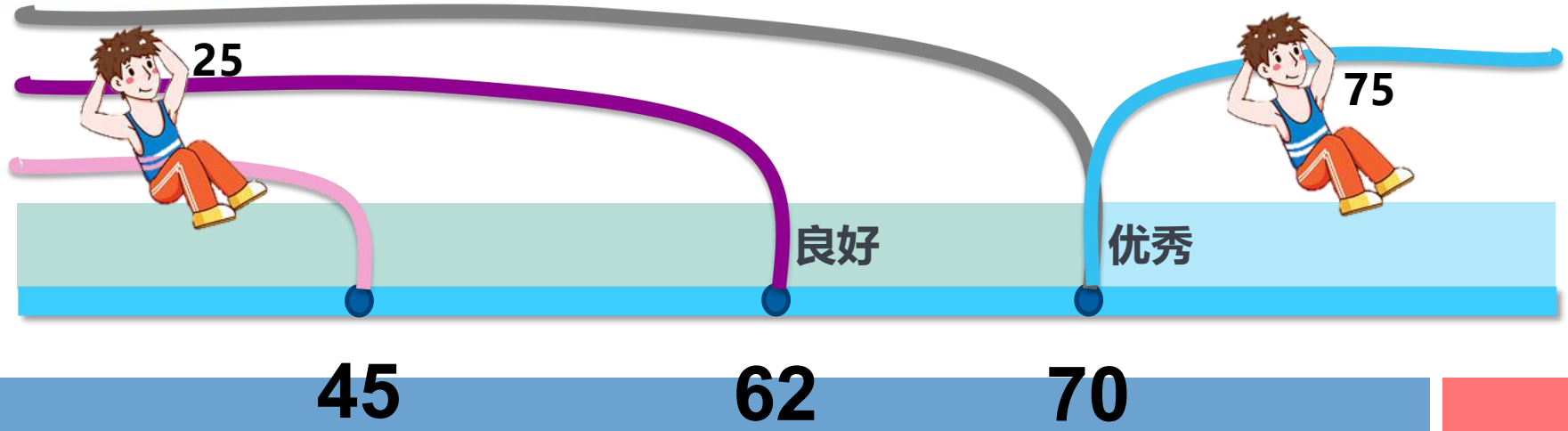
结论：这几个范围段之间是并列关系，可以在程序中调整顺序

2.4条件件表达式

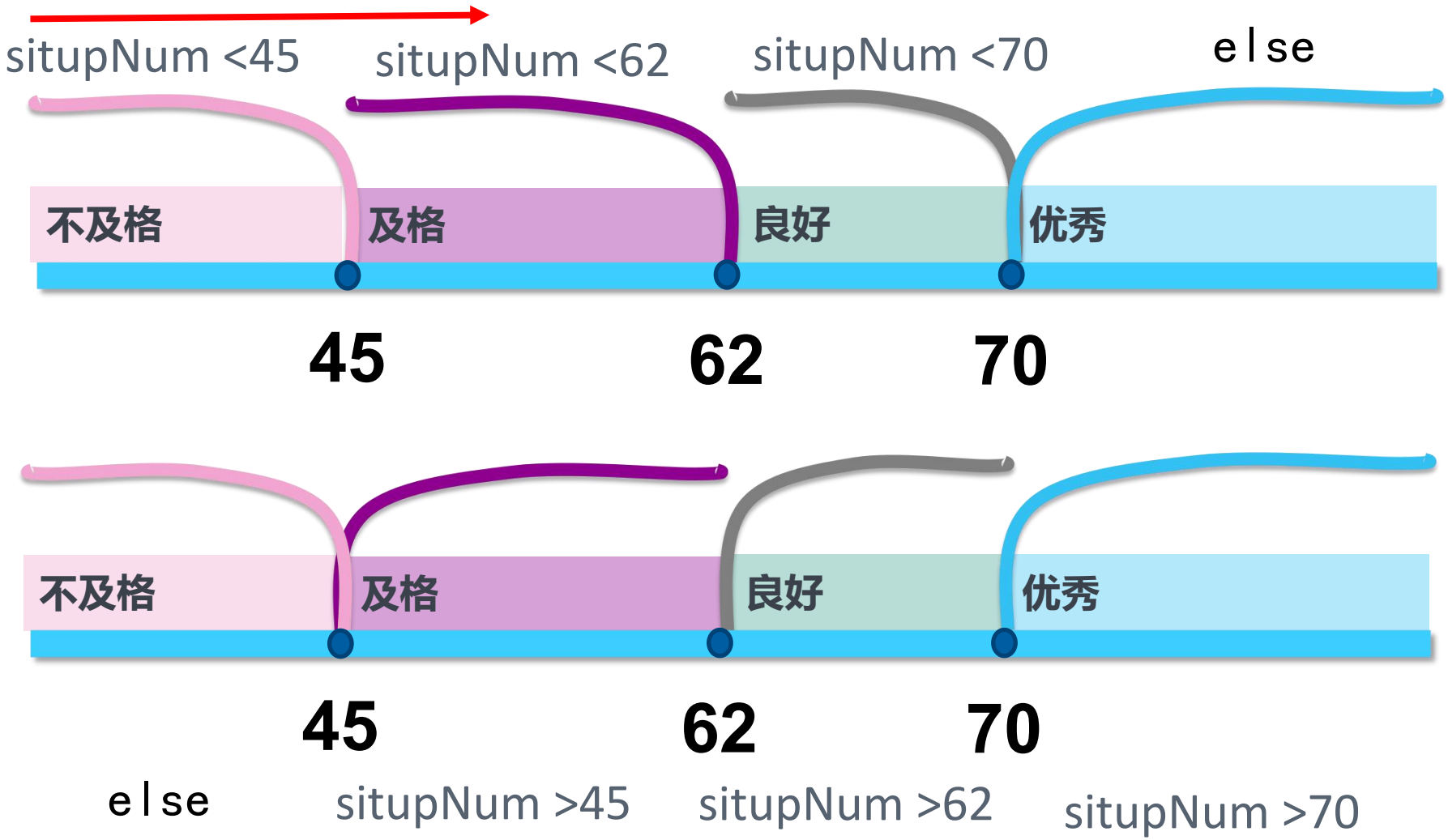
```
situpNum = int(input("仰卧起坐数量："))
if situpNum < 70:
    print("仰卧起坐考核为：良好")
elif situpNum < 62:
    print("仰卧起坐考核为：及格")
elif situpNum < 45:
    print("仰卧起坐考核为：不及格")
else:
    print("仰卧起坐考核为：优秀")
```

输入75，结果?
输入25，结果?

错误的条件包含关系



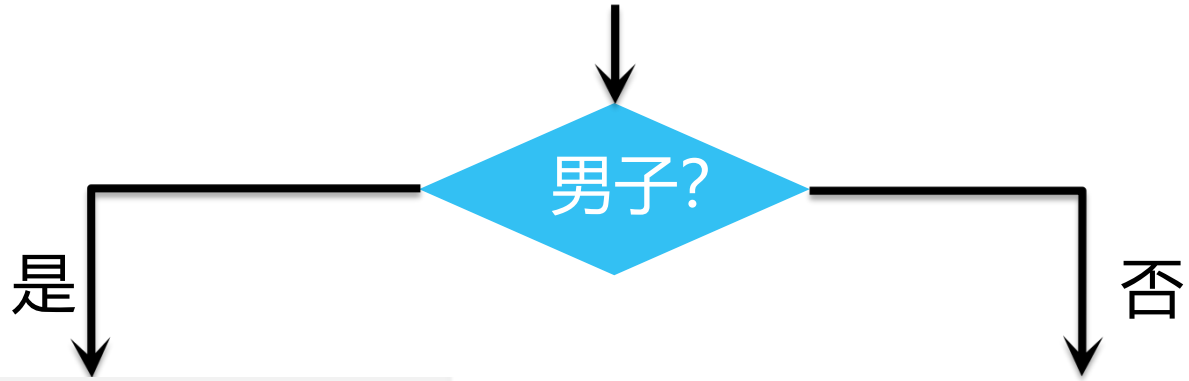
2.4条件件表达式




```
if situpNum < 45:
    grade="不及格"
elif situpNum < 62:
    grade = "及格"
elif situpNum < 70:
    grade = "良好"
else:
    grade="优秀"
```

```
if situpNum > 70:
    grade = "优秀"
elif situpNum > 62:
    grade = "良好"
elif situpNum >45:
    grade = "及格"
else:
    grade = "不及格"
```

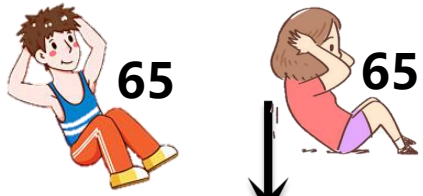
添加性别信息



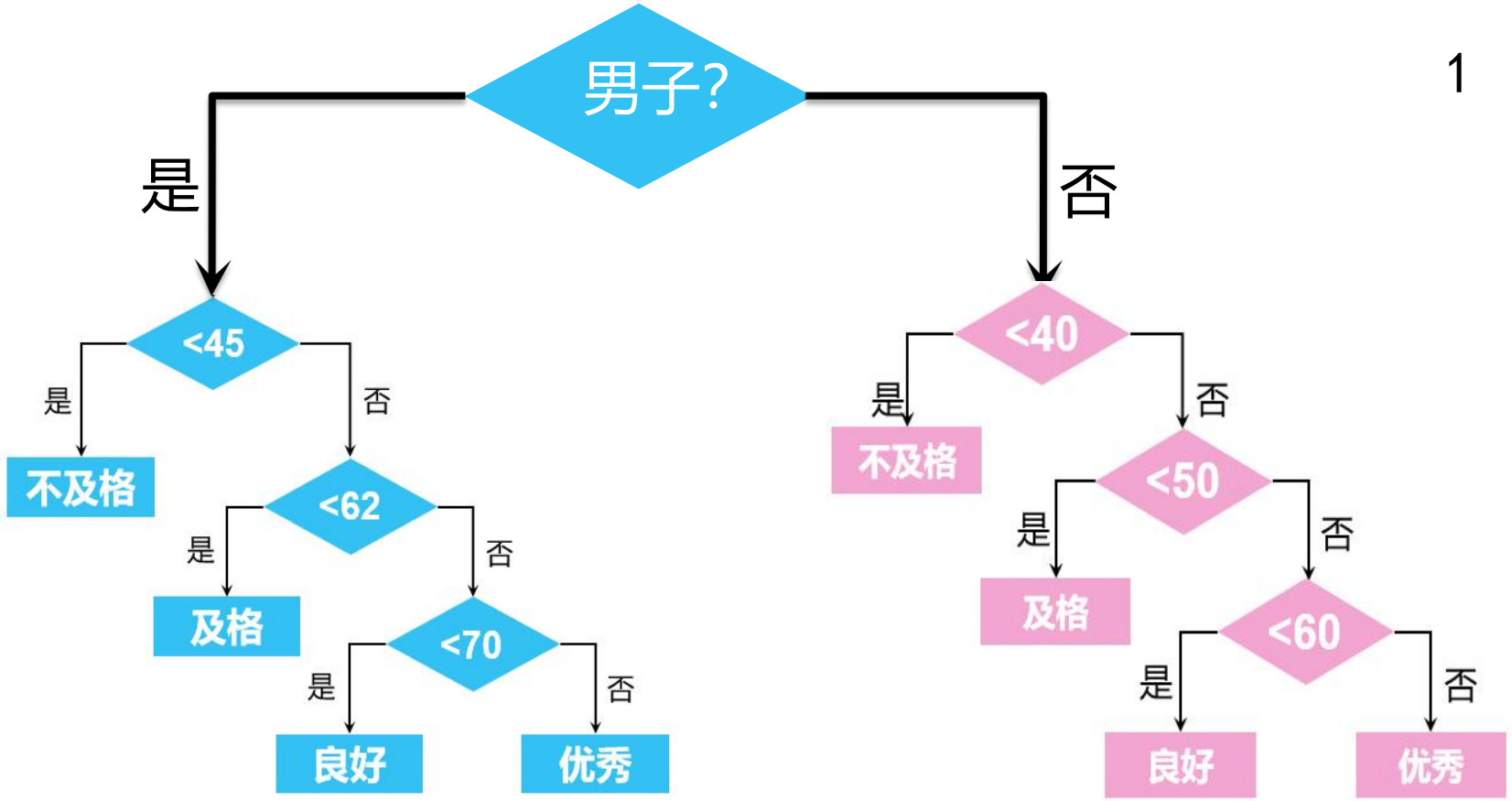

```
situpNum = int(input("仰卧起坐数量："))
if situpNum <45:
    print("仰卧起坐考核为：不及格")
elif situpNum < 62:
    print("仰卧起坐考核为：及格")
elif situpNum <70:
    print("仰卧起坐考核为：良好")
else:
    print("仰卧起坐考核为：优秀")
```

```
situpNum = int(input("仰卧起坐数量："))
if situpNum <40:
    print("仰卧起坐考核为：不及格")
elif situpNum < 50:
    print("仰卧起坐考核为：及格")
elif situpNum <60:
    print("仰卧起坐考核为：良好")
else:
    print("仰卧起坐考核为：优秀")
```

2.4条件件表达式



分支嵌套结构




```
if sex == "男" :
    situpNum = int(input("仰卧起坐数量："))
    if situpNum <45:
        print("仰卧起坐考核为：不及格")
    elif situpNum < 62:
        print("仰卧起坐考核为：及格")
    elif situpNum <70:
        print("仰卧起坐考核为：良好")
    else:
        print("仰卧起坐考核为：优秀")
else:
    situpNum = int(input("仰卧起坐数量："))
    if situpNum <40:
        print("仰卧起坐考核为：不及格")
    elif situpNum < 50:
        print("仰卧起坐考核为：及格")
    elif situpNum <60:
        print("仰卧起坐考核为：良好")
    else:
        print("仰卧起坐考核为：优秀")
```

1 2 3

注意：缩进必须要正确并且一致。

对比分析

```
score=75
if 90<=score<=100:
    print('A')
elif 80<=score<90:
    print('B')
elif 70<=score<80:
    print('C')
elif 60<=score<70:
    print('D')
elif 0<=score<60:
    print('E')
else:
    print('输入错误')
```

```
score=75
if 90<=score<=100:
    print('A')
if 80<=score<90:
    print('B')
if 70<=score<80:
    print('C')
if 60<=score<70:
    print('D')
if 0<=score<60 :
    print('E')
if score>100 or score<0:
    print('输入错误')
```

03

PYTHON分支语句
match语句

【案例分析】

解决问题3：如果是紧急通信(**URGENT**)，要立即上报；如果是常规通信(**CONFIRMED**)，10分钟内处理；如果是待处理通信(PENDING)，30分钟内处理；其他情况，标记为待核实，暂不处理。

"ALPHA7|N39-52.783,E116-25.467|T0830|CONFIRMED"

```
if status == "URGENT":
    print("立即上报")
elif status == "CONFIRMED":
    print("10分钟内处理")
elif status == "PENDING":
    print("30分钟内处理")
else:
    print("标记为待核实,暂不处理")
print("分类完成，等待下一条通信...")
```

```
# 使用match...case进行状态分类
match status:
    case "URGENT":
        print("立即上报")
    case "CONFIRMED":
        print("10分钟内处理")
    case "PENDING":
        print("30分钟内处理")
    case _:
        print("标记为待核实,暂不处理")
print("分类完成，等待下一条通信...")
```

- Python中match语句用来进行**匹配对象模式**进行分支结构，语法格式如下：

```
match 变量名:
    case 模式1:
        # 如果该变量匹配模式1，执行这里的代码
    case 模式2:
        # 如果该变量匹配模式2，执行这里的代码
    ...
    case _:
        # 如果该变量不匹配任何模式，执行这里的代码
```

通配符 _ 相当于else，缺省执行

具体值匹配（==）

```
m = int(input('序号是: '))

match m:
    case 1 | 3 | 5 | 7 | 9 | 10 | 12 | 14:
        print('执行紧急任务')
    case _:
        print('原地待命')
```

类型匹配

```
value = 1.1
match value:
    case int():
        print("这是一个整数")
    case float():
        print("这是一个浮点数")
    case str():
        print("这是一个字符串")
    case _:
        print("未知类型")
```

```
value = 1.1
if type(value) == int:
    print("这是一个整数")
elif type(value) == float:
    print("这是一个浮点数")
elif type(value) == str:
    print("这是一个字符串")
else:
    print("未知类型")
```

这是一个浮点数

分支结构在实际中的有哪些应用，你能举出例子吗？

【例】输入学号完成：根据学号信息打印输出学院、年级及专业信息。
请输入学号：5092025 1010
欢迎你，一院的2025级ZCQB专业学员！

【例】开发一个用户权限管理系统。系统中有三种用户角色："admin"、"user"和 "guest"。每个用户角色对应不同的权限级别。此外，系统还需要根据用户的年龄来进一步限制权限。具体规则如下：

- "admin"：拥有最高权限，无论年龄如何。
- "user"：拥有基本权限，但如果年龄小于 18 岁，则权限受限。
- "guest"：拥有最低权限，但如果年龄大于 60 岁，则权限受限。

AI目标检测结果后处理

AI目标检测是计算机视觉领域的核心技术，用于识别图像或视频中特定目标的位置和类别，输出目标类别、位置（边界框坐标）及检测置信度。

【题目描述】AI 目标检测（如自动驾驶、监控系统）中，需对检测结果进行后处理：筛选**置信度**达标的有效目标，判断目标**类别**，并校验**边界框**合法性。检测结果包含：**置信度（0.0-1.0，越高越可信）、边界框坐标（x,y,w,h，x/y 为左上角坐标，w/h 为宽高，均≥0）、类别 ID（1 = 行人，2 = 车辆，3 = 交通信号灯，其他为未知）**。规则：

1. 置信度不在 0.0-1.0 → 无效检测结果；
2. 边界框 x<0 或 y<0 或 w≤0 或 h≤0 → 边界框非法；
3. 类别 ID 对应合法类别，且置信度≥0.5 → 筛选通过，输出类别名称；
4. 类别 ID 合法但置信度 < 0.5 → 筛选未通过，提示置信度不足；
5. 类别 ID 非法 → 提示未知类别。

智能家居环境控制系统

智能家居 AI 系统根据环境数据（温度、湿度、光照度）和用户模式（1 = 节能模式，2 = 舒适模式），自动控制空调、加湿器、灯光。规则：

1、模式选择：区分节能 / 舒适模式/非法模式；

2、空调控制（温度 T，单位℃）：

节能模式：T<22 → 制热；22≤T≤28 → 关闭；T>28 → 制冷；

舒适模式：T<24 → 制热；24≤T≤26 → 关闭；T>26 → 制冷；

3、加湿器控制（湿度 H，单位 % RH）：

所有模式：H<40 → 开启；40≤H≤60 → 关闭；H>60 → 除湿；

4、灯光控制（光照度 L，单位 lux）：

节能模式：L<200 → 开启低亮度；200≤L≤500 → 关闭；L>500 → 关闭；

舒适模式：L<300 → 开启中亮度；300≤L≤600 → 关闭；L>600 → 关闭；

5、所有输入值需校验合法性（T：-10~45，H：0~100，L：0~2000，模式：1~2）。

总结

- 单分支结构 (if) - 基础条件判断
- 双分支结构 (if-else) - 二选一决策
- 多分支结构 (if-elif-else/match...case) - 多条件判断
- 嵌套选择结构 - 复杂逻辑处理

✓ 从条件到逻辑：不只是写if语句，而是构建完整的决策逻辑
✓ 从功能到健壮：考虑所有边界情况
✓ 从代码到系统：在更大系统中思考选择结构的作用
✓ 从技术到责任：理解代码在军事领域的应用中

下课并不代表思考的终止
期待我们下次的思想碰撞