人工智能程序设计

计算机学院

李晶晶

華南師範大學
South China Normal University

问题2

- 问题描述
  - 设计一个货币兑换程序，其功能是将美元、欧元等外币兑换成人民币。
  - 目前只考虑美元、欧元、日元兑换成人民币。

问题2

- 问题分析
  - 输入：带单位的外币金额（如：10USD、200EUR、3000JPY）
  - 处理：外币金额 * 相应的汇率（考虑输入错误处理）
    - ✓美元兑换人民币汇率：6.99
    - ✓欧元兑换人民币汇率：7.71
    - ✓日元兑换人民币汇率：0.065
  - 输出：带单位的人民币金额
  - 设计算法
    - ✓绘制流程图

涉及的基础知识二

- 字符串
- 集合类型
- 程序控制：分支语句

字符串

- Python中没有独立的字符数据类型，字符即长度为1的字符串
- 单引号（‘）、双引号（“ “）、三引号（’’’ 或者”””）
- 字符转义

  比如：’what’s happen’

  ➢转义符 \

  ’what\’s happen’　　#输出what’s happen

  ’good\\’　　#输出good\

字符串

```
s = 'a\tb\tc\\td'
print(s)      #输出：a    b    c\td
'\101'        #输出：'A'
'\x41'        #输出：'A'
```

- 字符转义
  - ➢转义字符

转义序列	字符	转义序列	字符
\'	单引号	\n	换行（LF）
\"	双引号	\r	
\\	反斜杠	\t	
\a	响铃（BEL）	\v	
\b	退格（BS）	\ooo	
\f	换页（FF）	\xhh	十六进制 Unicode 码对应的字符

```
s = 'd:\name\python'
print(s)
s = r'd:\name\python'
print(s)
```

  - ➢ 使用r’‘或R’ ’的字符串为原始字符串，其中任何字符都不进行转义

字符串

- 数值转化为字符串
  - 使用内置函数str()可以把数值转换为字符串
  - 使用print(123)输出数值时，将自动调用str(123)函数，把123转换为字符串，然后输出

字符串

- 字符串索引
  - 字符串是一种“序列”，可以用索引来获取元素的数据类型
  - 正索引、负索引

字符串

- 字符串基本操作
  - 提取指定位置的字符(方括号[ ])
  - 字符串切片
    - ✓截取字符串片段，形成子字符串
    - ✓[i:j] i表示截取字符串的开始索引，j代表结束索引
    - ✓半开闭区间：左闭右开
    - ✓省略第一个第一个索引，默认为0；省略第2个索引，默认为切片字符串的长度

```
word = 'python'
word[1]
word[-1]
```

```
word[0:19]
word[-1:3]
word[5:3]
```

```
word[0:3]
word[:3]
Word[3:]
word[-5:-1]
```

字符串

- 字符串基本操作
  - 字符串内容不可变
  - 指定位置的字符重新赋值，将会出错

```
word = 'python'
word[1] = 'p'
```

字符串

- 字符串基本操作
  - 字符串拼接
    - ✓用加号+将两个字符串拼接
    - ✓用星号*表示重复
  - 假设有字符串S=’abcde’
    - ✓增：在‘b’后增加‘z’

      A= S[:2] +'z'+ S[2:]
    - ✓删：从S中删除'b‘

      A=S[:1] + S[2:]
    - ✓改：字符串S中'b'改为'z‘

      A= S[:1] +'z'+ S[2:]

Python内置函数

- eval(<字符串>)函数是Python语言中一个十分重要的函数，它能够以Python表达式的方式解析并执行字符串，将返回结果输出

```
>>>x = 1
>>>eval("x + 1")
2
>>>eval("1.1 + 2.2")
3.3
```

集合类型

- 集合类型与数学中集合的概念一致，即包含0个或多个数据项的无序组合。
- 集合中元素不可重复。
- 集合元素类型只能是固定数据类型。
  - 例如：整数、浮点数、字符串、元组等。
  - 列表、字典和集合类型本身都是可变数据类型，不能作为集合的元素出现。

集合类型

- 集合是无序组合，没有索引和位置的概念，不能分片。
- 集合中元素可以动态增加或删除。
- 集合用大括号（{}）表示，可以用赋值语句生成一个集合。

```
>>>S = {425, "BIT", (10, "CS"), 424}
>>>S
{424, 425, (10, 'CS'), 'BIT'}
>>>T = {425, "BIT", (10, "CS"), 424, 425, "BIT"}
>>>T
{424, 425, (10, 'CS'), 'BIT'}
```

集合类型

- 集合类型主要用于三个场景：成员关系测试、元素去重和删除数据项。

```
>>>"BIT" in {"PYTHON", "BIT", 123, "GOOD"} #成员关系测试
True
>>>tup = ("PYTHON", "BIT", 123, "GOOD", 123) #元素去重
>>>set(tup)
{123, 'GOOD', 'BIT', 'PYTHON'}
>>>newtup = tuple(set(tup)–{'PYTHON'}) # 去重同时删除数据项
('GOOD', 123, 'BIT')
```

- 集合类型与其他类型最大的不同在于它不包含重复元素，因此，当需要对一维数据进行去重或进行数据重复处理时，一般通过集合来完成。

程序控制语句：分支语句

- 分支语句是控制程序运行的一类重要语句，它的作用是根据判断条件选择程序执行路径，其语法如下（注意缩进）：

```
if <条件1>:
    <语句块1>
elif <条件2>:
    <语句块2>
...
else:
    <语句块N>
```

问题2

- 问题描述
- 问题分析
- 涉及基础知识：字符串、集合及程序控制结构
- 程序实现
- 上机实践

问题2

- 演示代码
  - 输入：外币金额及单位（考虑多种外币）
    根据输入的字符串截取金额和币种
  - 处理：将输入的不同外币兑换成人民币
  - 输出：人民币金额或输入错误的处理
- 调试程序
  - 设断点
  - 单步跟踪

问题2

```python
# 汇率表
USD_VS_CNY = 6.99  # 美元兑换人民币
EUR_VS_CNY = 7.71  # 欧元兑换人民币
JPY_VS_CNY = 0.065  # 日元兑换人民币

# 带单位的外币输入
foreign_currency_str = input('请输入带单位的外币金额（如：100USD）：')

# 获取币种
currency = foreign_currency_str[-3:]
# 获取外币金额
f_str_value = foreign_currency_str[:-3]
f_c_value = eval(f_str_value)

cny_value = 0

if currency == 'USD':
    # 输入的是美元，计算兑换的人民币
    cny_value = f_c_value * USD_VS_CNY
elif currency == 'EUR':
    # 输入的是欧元，计算兑换的人民币
    cny_value = f_c_value * EUR_VS_CNY

elif currency == 'JPY':
    # 输入的是日元，计算兑换的人民币
    cny_value = f_c_value * JPY_VS_CNY

else:
    # 其他结果
    print('目前版本尚不支持该币种！')

# 输出结果
if currency in {'USD','EUR','JPY'}:
    print('兑换的人民币(CNY)金额是：', cny_value)
```

上机实践

- 温度转换
  - 输入：华氏或者摄氏温度值、温度标识（示例：20C、100F）
  - 处理：温度转化算法

    转化公式为：F = C * 1.8 + 32

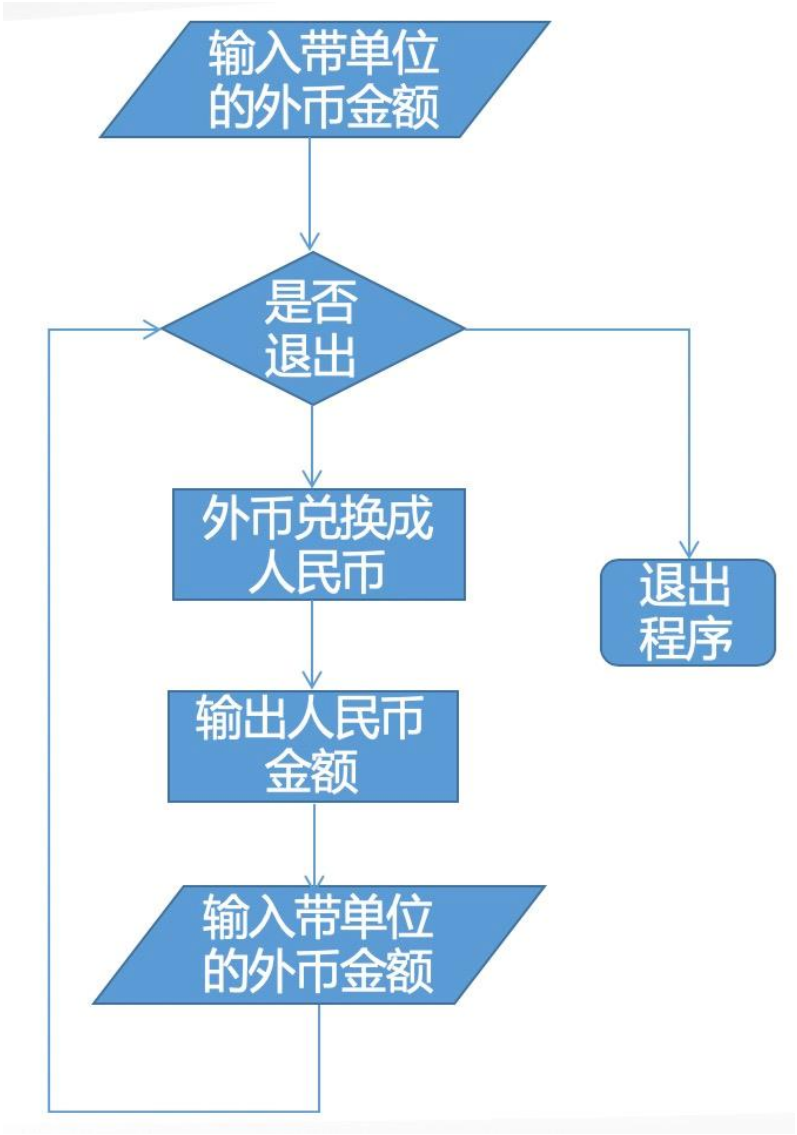
    C = (F – 32) / 1.8

    其中，C表示摄氏温度，F表示华氏温度
  - 输出：华氏或者摄氏温度值、温度标识

问题3

- 问题描述
  - 设计一个货币兑换程序，其功能是将美元、欧元等外币兑换成人民币。
  - 功能：判断将多种外币（目前只考虑美元、欧元、日元）兑换成人民币。
  - 增加功能：程序运行很多次，直到用户选择退出时才退出。

问题3



涉及的基础知识三

- 程序控制结构：循环语句

程序控制结构：循环语句

- 循环语句：控制程序运行，根据判断条件或计数条件确定一段程序的运行次数
  - for

```
for i in range(1, 6):
    print(i) #输出结果12345
```

  - for循环格式

1、无else子句格式：	2、带else子句格式：
for <控制变量> in <序列>: <循环体>	for <控制变量>in <序列>: <循环体> else: <语句块>

程序控制结构：循环语句

- for循环有两种基本结构：
  - for i in range()
  - for e in L

1、无else子句格式：	2、带else子句格式：
for <控制变量> in <序列>: 　　<循环体>	for <控制变量>in <序列>: 　　<循环体> else: 　　<语句块>

循环语句：for i in range()

- range()
  - 函数range(m, n, step)：起始值m（包括m），终止值 n（不包括n），步长step（相邻两个整数之间的间隔）。执行range(m, n, step)返回一个从m开始到n-1为止的步长为step的整数顺序。
    - ✓例如range(3,8,2)依次返回3,5,7；range(3,9,2)，依次产生3,5,7。
    - ✓注意，最后一个值必须要小于终止值，所以range(3,9,2)的最后值是7而不是9。

循环语句：for i in range()

- range()
  - 函数range(m,n)：省略step，则默认步长为1。
    - ✓例如range(2,5)则依次返回2,3,4。
  - 如果同时省略起始值m和步长step，即range中只有一个参数时，则会默认起始值为0，步长为1。
    - ✓例如range(5)将依次返回0,1,2,3,4。
  - 步长step可以为负数，但是当step为负数时，产生的最后一个值要大于终止值n。
    - ✓例如range(7,3,-2)依次返回7, 5。
    - ✓注意当step为负数时，最后迭代值要大于终止值，所以range(7,3,-2)最后迭代值不是3而是5。

循环语句：for e in L

- L为序列类型

```
#<例2.8：循环语句for e in L >
L = "hello"
for e in L:
    print(e, end=' ')
```

程序控制结构：循环语句

- 循环内的break与continue

```
# <例2.9：break>
L = 'hello world'
flag = True
for i in L:
    if i == ' ':
        flag = False; break
if flag:
    print("不存在空格")
else:
    print("存在空格")
```

```
# <例2.10: continue>
L = 'hello world'
for i in L:
    if i == ' ':
        continue
    print(i)
```

程序控制结构：循环语句

- while循环
  - ➢while循环格式

1、无else子句格式：

```
while<条件>:
    <循环体>
```

2、带else子句格式：

```
while<条件>:
    <循环体>
else:
    <else子句>
```

例2.9:判断一个

```
# <例2.11：判一个数是否为质数>
num = 7;a = num//2
while a>1:
    if num % a==0:
        print('num is not prime');break
    a = a - 1
else:    #没有执行break，则执行else
    print('num is a prime number')
```

问题3

- 演示代码
  - 程序不断运行，直到输入exit退出
  - 输入：外币金额及单位（考虑多种外币）或exit
    - ✓根据输入的字符串截取金额和币种
  - 处理：将输入的不同外币兑换成人民币或退出程序
  - 输出：人民币金额或输入错误的处理

问题3

```
# 汇率表
USD_VS_CNY = 6.99  # 美元兑换人民币
EUR_VS_CNY = 7.71  # 欧元兑换人民币
JPY_VS_CNY = 0.065  # 日元兑换人民币

# 带单位的外币输入
foreign_currency_str = input('请输入带单位的外

i = 0

while foreign_currency_str != 'exit':
    i = i + 1
    print('循环次数: ', i)
    # 获取币种
    currency = foreign_currency_str[-3:]
    # 获取外币金额
    f_str_value = foreign_currency_str[:-3]
    f_c_value = eval(f_str_value)

    cny_value = 0

    if currency == 'USD':
        # 输入的是美元，计算兑换的人民币
        cny_value = f_c_value * USD_VS_CNY
    elif currency == 'EUR':
        # 输入的是欧元，计算兑换的人民币
        cny_value = f_c_value * EUR_VS_CNY

    elif currency == 'JPY':
        # 输入的是日元，计算兑换的人民币
        cny_value = f_c_value * JPY_VS_CNY

    else:
        # 其他结果
        print('目前版本尚不支持该币种！')

    # 输出结果
    if currency in {'USD', 'EUR', 'JPY'}:
        print('兑换的人民币(CNY)金额是: ', cny_value)

    print('**************************************************')
    # 带单位的外币输入
    foreign_currency_str = input('请输入带单位的外币金额（退出程序输入exit）：')

print('程序已退出')
```

上机实践

- 温度转换（程序一直运行，直到输入exit退出）
  - 输入：华氏或者摄氏温度值、温度标识（示例：20C、100F）
  - 处理：温度转化算法

    转化公式为：$F = C * 1.8 + 32$

    $$C = (F - 32) / 1.8$$

    其中，C表示摄氏温度，F表示华氏温度
  - 输出：华氏或者摄氏温度值、温度标识

小结

- Python编码规范
- Python的输入和输出
- Python变量
- Python常用操作运算符
- 字符串
- 集合类型
- 程序控制：分支语句
- 程序控制结构：循环语句