

暨南大学本科实验报告专用纸

课程名称 python程序设计 成绩评定

实验项目名称 python作业第二章

姓名 崔嘉容 学号 2020100069 学院 网络空间安全 专业 网络空间安全

实验时间 2023 年 9 月 18 日 ~ 9 月 25 日 实验地点: 516

一、实验目的

Python程序由模块（即后缀为py的源文件）组成。模块包含语句，语句是python程序的基本构成元素。语句通常包含表达式，而表达式由操作数和运算符构成，用于创建和处理对象。通过本次实验掌握Python语言基础知识。

二、实验环境和设备

实验环境：操作系统-Windows10，python版本-3.11.3，开发环境-pycharm

实验设备：华为MateBook14-2020，处理器-i7-10510U，内存-16GB

三、实验内容和结果

题目一：编写程序，输入本金、年利率和年份，计算复利（保留一位小数）。

实验代码：

```
def compound_interest(principal, rate, years):
    if principal < 0 or rate < 0 or years < 0:
        raise ValueError("输入不能为负数")
    amount = principal * (1 + rate)**years
    return round(amount, 1)

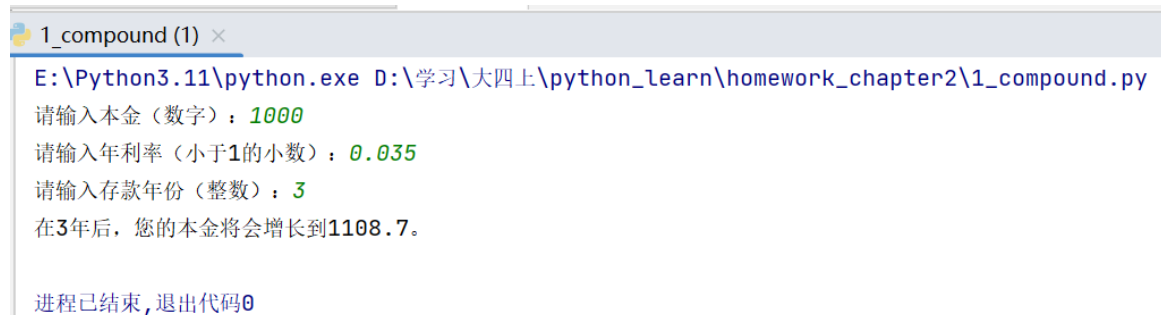
try:
    principal = float(input("请输入本金（数字）："))
    rate = float(input("请输入年利率（小于1的小数）："))
    if not (0 < rate < 1):
        raise ValueError("利率必须在0到1之间")
    years = int(input("请输入存款年份（整数）："))
```

暨南大学本科实验报告专用纸(附页)

```
result = compound_interest(principal, rate, years)

print(f"在{years}年后, 您的本金将会增长到{result}。")
except ValueError as e:
    print(f"发生错误: {e}")
except Exception as e:
    print(f"发生未知错误: {e}")
```

实验结果:



```
1_compound (1) x
E:\Python3.11\python.exe D:\学习\大四上\python_learn\homework_chapter2\1_compound.py
请输入本金(数字): 1000
请输入年利率(小于1的小数): 0.035
请输入存款年份(整数): 3
在3年后, 您的本金将会增长到1108.7。

进程已结束, 退出代码0
```

实验分析和总结:

1. 首先要保证最后数字保留1位小数, `round()` 函数确保了最终的计算结果保留了一位小数, `round()` 是Python内建的一个函数, 它用于四舍五入数字。
2. 对用户输入的合法性检查, 确保本金、年利率和存款年份均为非负数, 同时需要对利率进行限制确保其在0到1之间, 确保程序能够正确处理各种可能的输入。
3. 提升了对Python语言基础知识的理解, 特别是关于函数定义、异常处理和条件语句的应用。关于异常处理, 采用了两个关键异常块: `try` 块: 尝试执行可能会引发异常的代码。`except` 块: 如果在 `try` 块中引发了异常, 程序会转到相应的 `except` 块来处理异常, 可以通过指定特定类型的异常来处理。

```
try:
    # 可能会产生异常的代码块
except ValueError as e:
    # 当遇到 ValueError 异常时执行的代码块
except Exception as e:
    # 当遇到其他类型异常时执行的代码块
```

通过使用异常处理, 我们可以使程序在遇到意外情况时不会崩溃, 向用户提供有用的信息或采取必要的措施。

4. 通过编写一个实际问题的解决方案, 加深了对复利计算的理解, 并将其转化为可执行的程序。

暨南大学本科实验报告专用纸(附页)

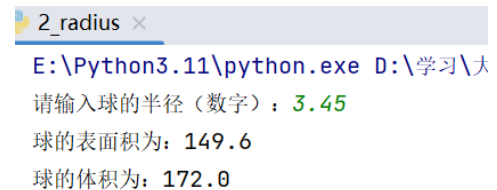
题目二: 编写程序, 输入半径, 计算球的表面积和体积 (保留一位小数) 。

实验代码:

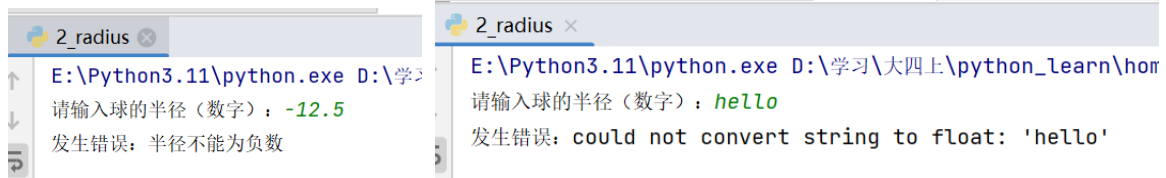
```
import math
def calculate_sphere_properties(radius):
    if radius < 0:
        raise ValueError("半径不能为负数")
    surface_area = round(4 * math.pi * radius ** 2, 1)
    volume = round((4 / 3) * math.pi * radius ** 3, 1)
    return surface_area, volume
try:
    radius = float(input("请输入球的半径 (数字): "))
    surface_area, volume = calculate_sphere_properties(radius)

    print(f"球的表面积为: {surface_area} ")
    print(f"球的体积为: {volume} ")
except ValueError as e:
    print(f"发生错误: {e}")
except Exception as e:
    print(f"发生未知错误: {e}")
```

实验结果:



2_radius x
E:\Python3.11\python.exe D:\学习\大
请输入球的半径 (数字): 3.45
球的表面积为: 149.6
球的体积为: 172.0



2_radius x
E:\Python3.11\python.exe D:\学>
请输入球的半径 (数字): -12.5
发生错误: 半径不能为负数

2_radius x
E:\Python3.11\python.exe D:\学习\大四上\python_learn\hor
请输入球的半径 (数字): hello
发生错误: could not convert string to float: 'hello'

实验分析和总结:

1. 计算球体的表面积和体积需要用到 π 值, 导入了 `math` 模块, 通过 `math.pi` 访问 π 的值。 π 的精度是有限的, `math.pi` 返回的是一个近似值, 如果需要更高精度, 可以考虑使用专门的数学库, 比如 `mpmath`。

2. 用户可能会输入一个非数字的字符串, 例如 "abc", 在程序中使用 `try...except` 块来捕获可能发生的 `ValueError` 异常, 并提供相应的错误消息。

3. 这个程序主要涉及数学计算, 而不涉及循环或递归等复杂算法, 因此时间复杂度和空间复杂度都相对较低。如果需要进一步优化, 可以考虑 ①优化精度, 但题目中仅要求保留一位小数; ② 可以考虑改进用户界面, 提供更友好的输入提示和结果显示。

暨南大学本科实验报告专用纸(附页)

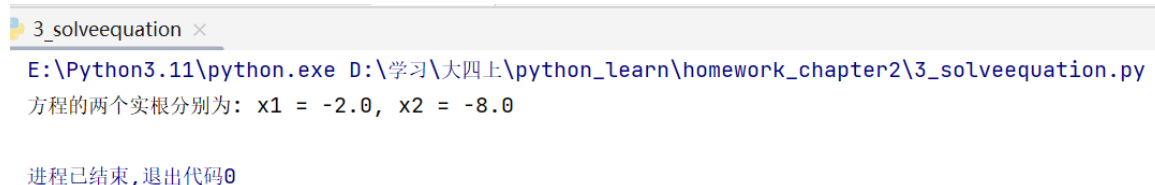
题目三：编写程序，求解一元二次方程 $x^2 + 10x + 16 = 0$ 。

实验代码：

```
import math
def solve_quadratic_equation(a, b, c):
    discriminant = b ** 2 - 4 * a * c

    if discriminant > 0:
        root1 = (-b + math.sqrt(discriminant)) / (2 * a)
        root2 = (-b - math.sqrt(discriminant)) / (2 * a)
        return root1, root2
    elif discriminant == 0:
        root = -b / (2 * a)
        return root,
    else:
        return None
# 给定方程的系数
a = 1
b = 10
c = 16
# 求解方程
solutions = solve_quadratic_equation(a, b, c)
# 输出结果
if solutions is not None:
    if len(solutions) == 2:
        print(f"方程的两个实根分别为: x1 = {solutions[0]}, x2 = {solutions[1]}")
    else:
        print(f"方程有一个实根: x = {solutions[0]}")
else:
    print("方程无实根")
```

实验结果：



```
3_solveequation x
E:\Python3.11\python.exe D:\学习\大四上\python_learn\homework_chapter2\3_solveequation.py
方程的两个实根分别为: x1 = -2.0, x2 = -8.0

进程已结束,退出代码0
```

实验分析和总结：

1. 可以先将整个程序分解成小部分，逐步编写并测试每个部分的功能。例

暨南大学本科实验报告专用纸(附页)

如，先编写一个函数来解一元二次方程，然后再编写用户输入和结果输出部分。

2. 在手算时通过因式分解可以是解一元二次方程的一种快速方法，特别是当方程容易因式分解时。但在计算机程序中，通常会使用更统一的公式法，因为它适用于所有类型的一元二次方程，无论是否容易因式分解。

3. `math.sqrt()` 是Python标准库中 `math` 模块提供的一个函数，用于计算一个数的平方根。

4. 在Python中，函数的返回值可以是任何类型，甚至可以是一个包含多种类型的元组。这使得Python中的函数非常灵活，可以根据需要返回不同类型的数据或者多个值。在求解方程的时候可以直接返回方程的一、两个实根或者无解的None形式。

5. 在Python中，变量的类型是动态的，不需要显式地指定。变量的类型会根据赋值的内容自动确定。在这里，`solutions` 的类型会根据 `solve_quadratic_equation(a, b, c)` 函数的返回值来自动确定。如果函数返回一个值，那么 `solutions` 将是该值的类型；如果函数返回 `None`，那么 `solutions` 将是 `NoneType` 类型。

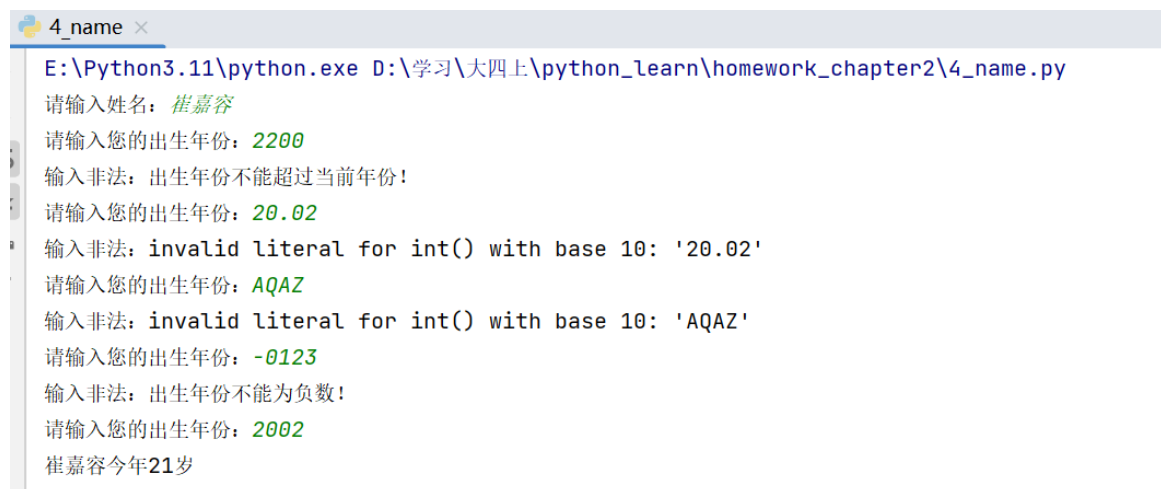
暨南大学本科实验报告专用纸(附页)

题目四: 编写程序, 输入姓名和出生年份, 输出姓名和年龄。

实验代码:

```
name = str(input("请输入姓名: "))
# 获取用户输入出生年份, 并添加合法性检查
while True:
    try:
        birth_year = int(input("请输入您的出生年份: "))
        if birth_year > 2023: # 出生年份不能超过当前年份
            raise ValueError("出生年份不能超过当前年份!")
        if birth_year < 0:
            raise ValueError("出生年份不能为负数!")
        break # 输入合法, 退出循环
    except ValueError as e:
        print(f"输入非法: {e}")
age = 2023 - birth_year
print(name + "今年" + str(age) + "岁")
```

实验结果:



```
E:\Python3.11\python.exe D:\学习\大四上\python_learn\homework_chapter2\4_name.py
请输入姓名: 崔嘉容
请输入您的出生年份: 2200
输入非法: 出生年份不能超过当前年份!
请输入您的出生年份: 20.02
输入非法: invalid literal for int() with base 10: '20.02'
请输入您的出生年份: AQAZ
输入非法: invalid literal for int() with base 10: 'AQAZ'
请输入您的出生年份: -0123
输入非法: 出生年份不能为负数!
请输入您的出生年份: 2002
崔嘉容今年21岁
```

实验分析和总结:

1. 在print输出的时候只能字符串之间拼接进行拼接, 不能把int类型进行拼接, “TypeError: can only concatenate str (not “int”) to str, 可以进行强制类型转换str(age), 进行拼接。

2. 想要输出错误的提示, print(ValueError + “输入无效, 请输入一个整数的出生年份。”), TypeError: unsupported operand type(s) for +: ‘type’ and ‘str’, 出现这个错误是因为将一个类型 (type) 和一个字符串 (str) 相加, 而这

暨南大学本科实验报告专用纸(附页)

两者是不兼容的。

采用`except ValueError as e: print(f"发生未知错误: {e}")`进行解决, 当程序执行 ``print(f"发生未知错误: {e}")`` 时:

``f"发生未知错误: {e}"``: 这是一个 f-string, 用于格式化字符串。在字符串中的 ``{e}`` 部分是一个表达式, 它会被替换为变量 ``e`` 的值; ``e``: 这是我们在 ``except`` 语句中定义的变量, 它保存了捕获到的异常对象; ``f-string`` 的将变量 ``e`` 的值插入到字符串中的 ``{e}`` 处; 最终的字符串将成为 ``f"发生未知错误: {e}"``, 其中 ``{e}`` 被替换为异常对象 ``e`` 的值。

在程序发生异常时, 能够将错误信息打印出来, 以使用户能够了解发生了什么问题。

3. 进一步对出生年份的限制应该不仅限制于非负整数, 还应该控制范围, 小于当前的年份, 在出生年份与当前年份相同的时候应该附加对月份的计算不然会出现“0岁”的结果, 同时应限制一个合理的年龄范围, 超出合理年龄范围的值应该再次提示用户进行重新检查。同时对当前年份的读取, 可以考虑读取系统时间来进行获取。

修改:

```
import datetime

def calculate_age(birth_year, birth_month):
    current_year = datetime.datetime.now().year
    current_month = datetime.datetime.now().month

    age_years = current_year - birth_year
    age_months = current_month - birth_month

    if age_months < 0:
        age_years -= 1
        age_months += 12

    return age_years, age_months

# 获取用户输入姓名
name = input("请输入您的姓名: ")

# 获取用户输入出生年份和月份, 并添加合法性检查
while True:
    try:
```

暨南大学本科实验报告专用纸(附页)

```
birth_year = int(input("请输入您的出生年份："))
birth_month = int(input("请输入您的出生月份："))
current_year = datetime.datetime.now().year

if birth_year > current_year or birth_year < 0:
    raise ValueError("出生年份无效，请重新输入。")

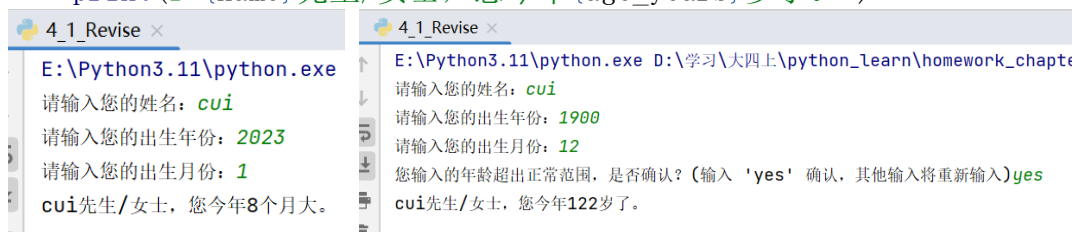
if birth_month > 12 or birth_month < 1:
    raise ValueError("出生月份无效，请重新输入。")

if birth_year == current_year and birth_month >
datetime.datetime.now().month:
    raise ValueError("出生月份无效，请重新输入。")

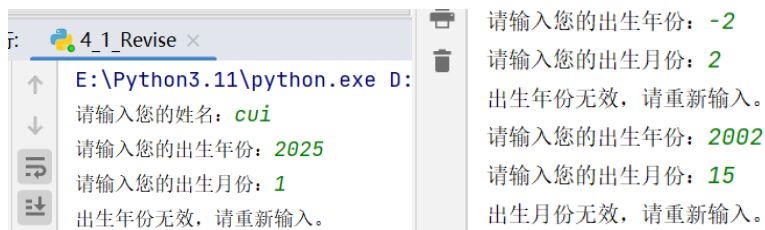
if current_year - birth_year > 120: # 设置一个合理的年龄范围
    (例如，120岁)
    confirm = input("您输入的年龄超出正常范围，是否确认？(输入
'yes' 确认，其他输入将重新输入)")
    if confirm.lower() != 'yes':
        continue

    break # 输入合法，退出循环
except ValueError as e:
    print(e)

# 计算年龄
age_years, age_months = calculate_age(birth_year, birth_month)
# 输出姓名和年龄
if age_years == 0:
    print(f"{name}先生/女士，您今年{age_months}个月大。")
else:
    print(f"{name}先生/女士，您今年{age_years}岁了。")
```



The screenshot shows a terminal window titled '4_1_Revise'. The command prompt is 'E:\Python3.11\python.exe'. The user enters 'cui' for the name, '2023' for the birth year, and '1' for the birth month. The program outputs 'cui先生/女士，您今年8个月大。'.



The screenshot shows a terminal window titled '4_1_Revise'. The command prompt is 'E:\Python3.11\python.exe'. The user enters 'cui' for the name, '2025' for the birth year, and '1' for the birth month. The program outputs '出生年份无效，请重新输入。'.