**实验报告**

|  |  |  |
| --- | --- | --- |
| 课程名称：python程序设计实验课 | 班级：软件1912 | 实验日期：2021-4-29 |
| 姓名：李越 | 学号：20192071 | 指导老师：杨帆 |
| 实验名称：循环语句 | 实验序号：09 | 实验成绩： |
| 1. 实验目的： 2. 动态显示数字时钟 3. 键盘事件响应函数（贴瓷砖 ） 4. 编写一个程序，提示用户输入以恶搞整数，然后反向显示一个整数 5. 猴子吃桃问题：猴子第一天摘下若干个桃子，当即吃了一半，还不瘾，又多吃了一个第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第10天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。   二、实验环境：  Python3.9.2 win10  三、实验步骤：  1.   1. """ 2. 功能：动态显示数字时钟 3. 日期：2021-04-29 4. 作者：1912班 李越 5. """ 7. **from** turtle **import** \* 8. **from** datetime **import** \*  11. # 移动到指定位置 12. **def** skip(step): 13. penup() 14. forward(step) 15. pendown()  18. # 画指针 19. **def** drawpointer(name, length): 20. reset() 21. skip(-length \* 0.1) 22. begin\_poly() 23. forward(length \* 1.1) 24. end\_poly() 25. handForm = get\_poly() 26. register\_shape(name, handForm)  29. # 初始化 30. **def** init(): 31. **global** hrpointer, minpointer, secpointer, weektext 32. # 设置turtle Logo模式，朝北 33. mode("logo") 34. drawpointer("hrpointer", 90) 35. drawpointer("minpointer", 130) 36. drawpointer("secpointer", 140) 37. hrpointer = Turtle() 38. hrpointer.shape("hrpointer") 39. minpointer = Turtle() 40. minpointer.shape("minpointer") 41. secpointer = Turtle() 42. secpointer.shape("secpointer") 43. secpointer.pencolor("red") 44. **for** pointer **in** hrpointer, minpointer, secpointer: 45. pointer.shapesize(3, 1, 1) 46. pointer.speed(0) 47. # 文字输出 48. weektext = Turtle() 49. weektext.hideturtle() 50. weektext.penup()  53. # 设置表盘形状 54. **def** setupClock(radius): 55. reset() 56. #  clockPanel = Turtle() 57. pensize(7) 58. pencolor("blue") 59. **for** i **in** range(60): 60. skip(radius) 61. **if** i % 5 == 0: 62. forward(20) 63. skip(-radius - 20) 64. **else**: 65. dot(5) 66. skip(-radius) 67. right(6)  70. # 星期文本 71. **def** Week(t): 72. week = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"] 73. **return** week[t.weekday()]  76. # 日期文本 77. **def** Date(t): 78. y = t.year 79. m = t.month 80. d = t.day 81. **return** "%s %d %d" % (y, m, d)  84. # 绘制表针动态显示 85. **def** tick(): 86. t = datetime.today() 87. second = t.second + t.microsecond \* 0.0000001 88. minute = t.minute + second / 60.0 89. hour = t.hour + minute / 60.0 90. secpointer.setheading(6 \* second) 91. minpointer.setheading(6 \* minute) 92. hrpointer.setheading(30 \* hour) 93. tracer(False) 94. weektext.forward(65) 95. weektext.write(Week(t), align="center", font=("Courier", 14, "bold")) 96. weektext.back(130) 97. weektext.write(Date(t), align="center", font=("Courier", 14, "bold")) 98. weektext.home() 99. tracer(True) 100. # 间隔100ms调用一次 101. ontimer(tick, 100)  104. **def** main(): 105. tracer(False) 106. init() 107. setupClock(160) 108. tracer(True) 109. tick() 110. mainloop()  113. **if** \_\_name\_\_ == '\_\_main\_\_': 114. main() 116. secpointer.shape("secpointer") 117. secpointer.pencolor("red") 118. **for** pointer **in** hrpointer, minpointer, secpointer: 119. pointer.shapesize(3, 1, 1) 120. pointer.speed(0) 121. # 文字输出 122. weektext = Turtle() 123. weektext.hideturtle() 124. weektext.penup()  127. # 设置表盘形状 128. **def** setupClock(radius): 129. reset() 130. #  clockPanel = Turtle() 131. pensize(7) 132. pencolor("blue") 133. **for** i **in** range(60): 134. skip(radius) 135. **if** i % 5 == 0: 136. forward(20) 137. skip(-radius - 20) 138. **else**: 139. dot(5) 140. skip(-radius) 141. right(6)  144. # 星期文本 145. **def** Week(t): 146. week = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"] 147. **return** week[t.weekday()]  150. # 日期文本 151. **def** Date(t): 152. y = t.year 153. m = t.month 154. d = t.day 155. **return** "%s %d %d" % (y, m, d)  158. # 绘制表针动态显示 159. **def** tick(): 160. t = datetime.today() 161. second = t.second + t.microsecond \* 0.0000001 162. minute = t.minute + second / 60.0 163. hour = t.hour + minute / 60.0 164. secpointer.setheading(6 \* second) 165. minpointer.setheading(6 \* minute) 166. hrpointer.setheading(30 \* hour) 167. tracer(False) 168. weektext.forward(65) 169. weektext.write(Week(t), align="center", font=("Courier", 14, "bold")) 170. weektext.back(130) 171. weektext.write(Date(t), align="center", font=("Courier", 14, "bold")) 172. weektext.home() 173. tracer(True) 174. # 间隔100ms调用一次 175. ontimer(tick, 100)  178. **def** main(): 179. tracer(False) 180. init() 181. setupClock(160) 182. tracer(True) 183. tick() 184. mainloop()  187. **if** \_\_name\_\_ == '\_\_main\_\_': 188. main()   2.   1. """ 2. 功能：键盘事件响应函数（贴瓷砖） 3. 日期：2021-04-29 4. 作者：1912班 李越 5. """ 7. **import** turtle 9. unit\_length = 100 10. width = 4 11. height = 4 12. origin\_x = 0  # 由于移动瓷砖需要修改原点坐标，因此使用变量，而不使用元组 13. origin\_y = 0 14. rotate = 0  # 旋转次数 15. edge\_point\_list = [] 16. center\_point\_list = []  19. **def** draw\_L(): 20. edge\_point\_list.clear() 21. turtle.reset()  # turtle 清除窗口 22. turtle.penup() 23. turtle.goto(origin\_x, origin\_y)  # 光标移动到原点坐标 24. turtle.right(90 \* (rotate % 4))  # 根据旋转次数计算旋转角度 25. turtle.pendown() 26. turtle.fillcolor("blue") 27. turtle.begin\_fill() 28. turtle.forward(100) 29. turtle.right(90) 30. turtle.forward(100) 31. edge\_point\_list.append(turtle.position()) 32. turtle.right(90) 33. turtle.forward(100 \* 2) 34. edge\_point\_list.append(turtle.position()) 35. turtle.right(90) 36. turtle.forward(100 \* 2) 37. edge\_point\_list.append(turtle.position()) 38. turtle.right(90) 39. turtle.forward(100) 40. turtle.right(90) 41. turtle.forward(100) 42. turtle.end\_fill() 43. center\_point\_list.clear() 44. center\_point\_list.append(((origin\_x + edge\_point\_list[0][0]) \* 0.5, (origin\_y + edge\_point\_list[0][1]) \* 45. 0.5)) 47. center\_point\_list.append(((origin\_x + edge\_point\_list[1][0]) \* 0.5, (origin\_y + edge\_point\_list[1][1]) \* 48. 0.5)) 50. center\_point\_list.append(((origin\_x + edge\_point\_list[2][0]) \* 0.5, (origin\_y + edge\_point\_list[2][1]) \* 51. 0.5)) 52. **print**(edge\_point\_list) 53. **print**(center\_point\_list)  56. **def** move\_up\_tiling(): 57. **global** origin\_x 58. **global** origin\_y 59. origin\_x = origin\_x 60. origin\_y = origin\_y + 1 \* unit\_length 61. draw\_L()  64. **def** move\_down\_tiling(): 65. **global** origin\_x 66. **global** origin\_y 67. origin\_x = origin\_x 68. origin\_y = origin\_y - 1 \* unit\_length 69. draw\_L()  72. **def** move\_left\_tiling(): 73. **global** origin\_x 74. **global** origin\_y 75. origin\_x = origin\_x - 1 \* unit\_length 76. origin\_y = origin\_y 77. draw\_L()  80. **def** move\_right\_tiling(): 81. **global** origin\_x 82. **global** origin\_y 83. origin\_x = origin\_x + 1 \* unit\_length 84. origin\_y = origin\_y 85. draw\_L()  88. **def** rotate\_tiling(): 89. **global** rotate 90. rotate = rotate + 1 91. draw\_L() 92. turtle.setup((width + 5) \* unit\_length, (height + 2) \* unit\_length) 94. **def** main(): 95. win = turtle.Screen() 96. win.tracer(0)  # 不显示绘制轨迹 97. win.onkey(draw\_L, 't')  # 按T键绘制瓷砖 98. win.onkey(move\_up\_tiling, 'Up')  # 按向上键，向上移动 99. win.onkey(move\_down\_tiling, 'Down')  # 按向下键，向下移动 100. win.onkey(move\_left\_tiling, 'Left')  # 按向左键，向左移动 101. win.onkey(move\_right\_tiling, 'Right')  # 按向右键，向右移动 102. win.onkey(rotate\_tiling, 'r')  # 按 R 键，顺时针旋转 90 度 103. win.listen()  # 窗口监听 104. win.mainloop() 106. **if** \_\_name\_\_ == '\_\_main\_\_': 107. main() 109. secpointer.shape("secpointer") 110. secpointer.pencolor("red") 111. **for** pointer **in** hrpointer, minpointer, secpointer: 112. pointer.shapesize(3, 1, 1) 113. pointer.speed(0) 114. # 文字输出 115. weektext = Turtle() 116. weektext.hideturtle() 117. weektext.penup()  120. # 设置表盘形状 121. **def** setupClock(radius): 122. reset() 123. #  clockPanel = Turtle() 124. pensize(7) 125. pencolor("blue") 126. **for** i **in** range(60): 127. skip(radius) 128. **if** i % 5 == 0: 129. forward(20) 130. skip(-radius - 20) 131. **else**: 132. dot(5) 133. skip(-radius) 134. right(6)  137. # 星期文本 138. **def** Week(t): 139. week = ["星期一", "星期二", "星期三", "星期四", "星期五", "星期六", "星期日"] 140. **return** week[t.weekday()]  143. # 日期文本 144. **def** Date(t): 145. y = t.year 146. m = t.month 147. d = t.day 148. **return** "%s %d %d" % (y, m, d)  151. # 绘制表针动态显示 152. **def** tick(): 153. t = datetime.today() 154. second = t.second + t.microsecond \* 0.0000001 155. minute = t.minute + second / 60.0 156. hour = t.hour + minute / 60.0 157. secpointer.setheading(6 \* second) 158. minpointer.setheading(6 \* minute) 159. hrpointer.setheading(30 \* hour) 160. tracer(False) 161. weektext.forward(65) 162. weektext.write(Week(t), align="center", font=("Courier", 14, "bold")) 163. weektext.back(130) 164. weektext.write(Date(t), align="center", font=("Courier", 14, "bold")) 165. weektext.home() 166. tracer(True) 167. # 间隔100ms调用一次 168. ontimer(tick, 100)  171. **def** main(): 172. tracer(False) 173. init() 174. setupClock(160) 175. tracer(True) 176. tick() 177. mainloop()  180. **if** \_\_name\_\_ == '\_\_main\_\_': 181. main()   3.   1. """ 2. 功能：反向输出整数 3. 日期：2021-04-29 4. 作者：1912班 李越 5. """  8. **def** reserve(num): 9. i = 0 10. num1 = num 11. **while** True: 12. **if** num1 // 10 == 0: 13. **break** 14. i += 1 15. num1 = num1 // 10 16. sum = 0 17. **while** i >= 0: 18. sum = sum + (num % 10) \* (10 \*\* i) 19. num = num // 10 20. i = i - 1 21. **return** sum  24. **def** main(): 25. num = int(input("请输入一个数:")) 26. **print**(reserve(num))  29. **if** \_\_name\_\_ == '\_\_main\_\_': 30. main()   4.   1. """ 2. 功能：猴子吃桃 3. 日期：2021-04-29 4. 作者：1912班 李越 5. """  8. **def** monkey\_tao(n): 9. **if** n > 10 **or** n < 1: 10. **return** 11. **elif** n == 10: 12. **return** 1 13. **else**: 14. **return** (monkey\_tao(n + 1) + 1) \* 2  17. **def** main(): 18. **print**(monkey\_tao(1))  21. **if** \_\_name\_\_ == '\_\_main\_\_': 22. main()   四、实验结果：  1.    2.    3.    4.    五、实验心得：  人生苦短我用python. | | |