第一题

**题目：**

定义一个存储数字的class KNumber。

输出 KNumber 对象时，以韩语发音输出。

例）输入：5

输出：오

输入：-4

输出：마이너스 사

**输出举例：**

-10：마이너스 십

-9 ：마이너스 구

-8 ：마이너스 팔

-7: 마이너스 칠

-6:마이너스 육

-5:마이너스 오

-4:마이너스 사

-3:마이너스 삼

-2:마이너스 이

-1:마이너스 일

0:영

1:일

2:이

3:삼

4:사

5:오

6:육

7:칠

8:팔

9:구

10: 십

**提示：**

class KNumber:

#class constructor

# 将数字作为参数输入并保存为属性(Noedit)

def \_\_init\_\_(self, num):

self.number = num

# 转换字符串的魔法宣言

# - 数字用韩语发音返回

# - 负数的话，添加在前面添加"마이너스"

（注释：这里的： '마이너스'意思是 ‘-’ 负号）

def? # (Edit 1) 提示 - 用于将对象转换为字符串的 Magic 方法

num\_names = '영일이삼사오육칠팔구십' # (No edit)

（注释：这里的： '영일이삼사오육칠팔구십'意思是‘0，1，2，3，4，5，6，7，8，9，10’）

if True: # 需要修改 - (Edit 2) 提示 - 如果是负数

return ‘마이너스’（负号） + ‘이 문자열 지우고 수정’（删除并修改此字符串） # (Edit 3)

elif True: # 需要修改 - (Edit 4) 提示 - 如果是正数

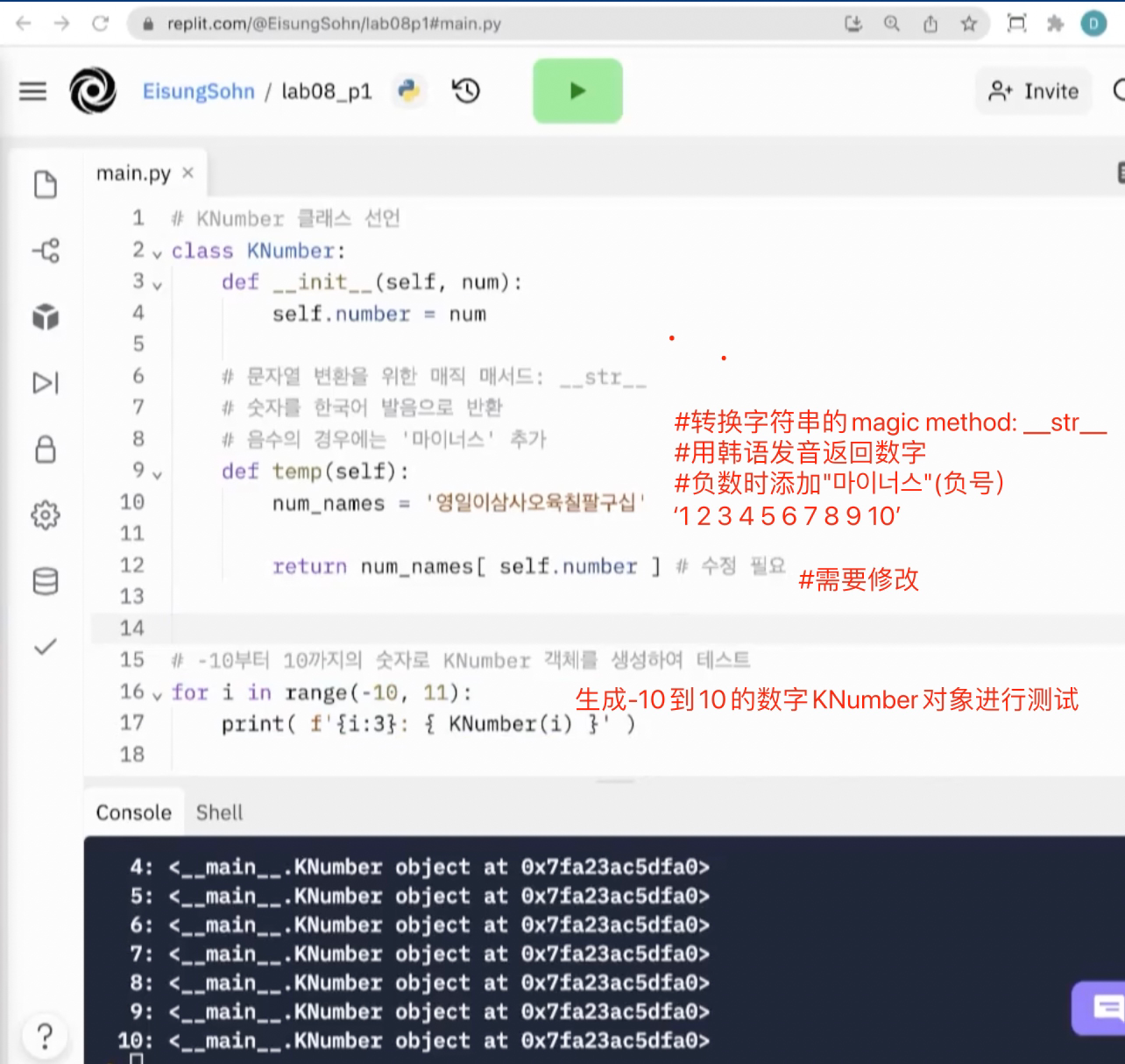
return num\_names[ self.number ] # (No edit)

# DO NOT EDIT

# 以 -10 到 10 的数字创建 KNumber 对象并进行测试( Noedit)

for i in range(-10, 11):

print( f'{i:3}: { KNumber(i) }' )



第二题

**题目：**

定义一种银行用户 class User

magic method( 创建者) \_ \_ init \_ \_: 重置为名称和余额

send\_to() : 汇款method

参数:收件人、金额

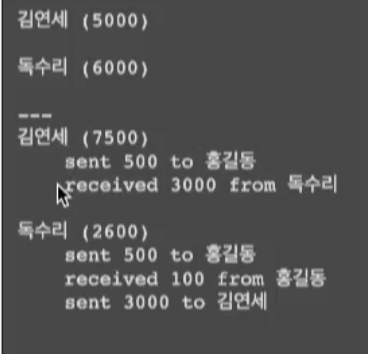
receive\_from():汇款method

参数:发送人，金额

magic method \_ \_ str \_:

输出余额和存取款记录

**输出举例：**



ps：韩文部分均为人名

**提示：**

# class User:

# class constructor( No edit)

def \_\_init\_\_(self, name, balance):

self.name = name

self.balance = balance

self.history = []

# 汇款方式

def send\_to(self, recipient, amount):

if True: # 将条件修改为"잔고가 보낼 금액보다 크다면“（如果余额大于要发送的金额）(Edit 1)

# 调用 User 对象 receive\_from() 方法( Edit 2)

# 提示) 从recipient.receive\_from开始

# 修改余额( Edit 3)

# 提示) 需要从self. balance 中删除 amount

self.history.append(f'sent{amount} to {recipient.name}') # (No edit)

# 汇款方法

def receive\_from(self, sender, amount):

# 修改余额( Edit 4)

# 提示) 通过仅更改汇款方式中修改余额方式的符号

self.history.append(f'received {amount} from {sender.name}') # (No edit)

# 输出余额# (Noedit)

def \_\_str\_\_(self):

result = f'{self.name} ({self.balance})\n'

for item in self.history:

result += '\t' + item + '\n'

return result

# 测试代码( Noedit)

kim = User('김연세', 5000)

hong = User('홍길동', 2000)

eagle = User('독수리', 6000)

print(kim)

print(eagle)

print('---')

kim.send\_to( hong, 500 )

eagle.send\_to( hong, 500 )

hong.send\_to( eagle, 100 )

eagle.send\_to( kim, 3000 )

print(kim)

print(eagle)

