

LU12.L03 - Lotto

authenticate.py

authenticate.py

```
from person import Person

def login():
    people = load_people()
    password = input('Passwort > ')
    for person in people:
        if password == person.password:
            return person

    return None

def load_people():
    """
    loads the list of people
    :return: list of person-objects
    """
    people_list = list()
    people_list.append(Person('Inga', 'geheim', 14.00))
    people_list.append(Person('Peter', 'secreät', 7.00))
    people_list.append(Person('Beatrice', 'passWORT', 23.00))
    return people_list

if __name__ == '__main__':
    person = login()
    assert person is None
```

lottery.py

lottery.py

```
from numeric_input import read_int
from ticket import Ticket

def create_ticket(person):
```

```
if person.balance >= 2.00:
    person.balance -= 2.00
    ticket = Ticket(list(), 0)
    select_numbers(ticket)
    print_ticket(ticket)
    print(f'Guthaben: {person.balance}')
else:
    print('Dein Guthaben reicht nicht aus')

def select_numbers(ticket):
    count = 0
    while count < 5:
        number = read_int('Lottozahl', 1,42)
        if number in ticket.numbers:
            print ('Diese Zahl wurde schon gewählt')
        else:
            ticket.numbers.append(number)
            count+=1

    number = read_int('Jokerzahl', 1,6)
    ticket.joker = number

def print_ticket(ticket):
    for i in range(1,42):
        if i in ticket.numbers:
            print (' X ', '')
        else:
            print (i)
        if i % 6 == 0:
            print(' ')

    print (ticket.joker)

if __name__ == '__main__':
    pass
```

main.py

main.py

```
from authenticate import login
from lottery import create_ticket
from menu import select_menu
from money import transfer_money
```

```
def main():
    person = login()
    selection = ' '
    while person is not None and selection not in 'Zz':
        selection = select_menu()
        if selection in 'Aa':
            transfer_money(person)
        elif selection in 'Bb':
            create_ticket(person)

if __name__ == '__main__':
    main()
```

menu.py

menu.py

```
def show_menu():
    print('Lotto\n' +
          '-----\n' +
          'A) Konto Ein- und Auszahlungen tätigen\n' +
          'B) Lottotipps abgeben\n' +
          'Z) Beenden')

def select_menu():
    choice = None
    while choice is None:
        choice = input('Auswahl (A/B/C/Z) > ')
        if choice not in 'AaBbCcZz':
            print('Ungültige Auswahl')
            choice = None

    return choice

if __name__ == '__main__':
    pass
```

money.py

money.py

```
from numeric_input import read_float

def transfer_money(person):
    choice = ' '
    while choice not in 'Zz':
        print(f'Guthaben: {person.balance}')
        choice = input('Auswahl > ')
        if choice in 'Aa':
            amount = read_float('Auszahlung', 10, 20)
            person.balance -= amount
        if choice in 'Ee':
            amount = read_float('Einzahlung', 10, 20)
            person.balance += amount

def select_transaction():
    transaction = None
    while transaction is None:
        transaction = input('Auswahl (A/E/Z) >')
        if transaction not in ['A', 'a', 'E', 'e', 'Z', 'z']:
            transaction = None
        print('Keine gültige Auswahl')

    return transaction

if __name__ == '__main__':
    pass
```

numeric_input.py

numeric_input.py

```
def read_int(prompt, min_number, max_number):
    number = None
    while number is None:
        try:
            number = int(input(f'{prompt} >'))
            if min_number < number > max_number:
                print(f'Gib eine Zahl von {min_number} bis {max_number} ein')
```

```

        number = None
    except ValueError:
        print (f'Keine gültige Lottozahl')

    return number

def read_float(prompt, min_number, max_number):
    number = None
    while number is None:
        try:
            number = float(input(f'{prompt} >'))
            if min_number < number > max_number:
                print (f'Gib eine Zahl von {min_number} bis
{max_number} ein')
            number = None
        except ValueError:
            print (f'Keine gültige Zahl')

    return number

if __name__ == '__main__':
    pass

```

person.py

person.py

```

from dataclasses import dataclass

@dataclass
class Person:
    """
    A person playing the lottery
    """
    name: str
    password: str
    balance: float

if __name__ == '__main__':
    # used for testing
    person = Person('Hans', 'test', 25.75)
    assert person.name == 'Hans'
    assert person.password == 'test'

```

```
assert person.balance == 25.75
```

ticket.py

ticket.py

```
from dataclasses import dataclass

@dataclass
class Ticket:
    """
    A lottery ticket with a list of numbers and the joker
    """
    numbers: list
    joker: int

if __name__ == '__main__':
    my_numbers = [1,17,8,32]
    ticket = Ticket(my_numbers, 7)
    assert 17 in ticket.numbers
    assert ticket.joker == 7
```

m319-AnG

✖ Marcel Suter

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/de/modul/m319/learningunits/lu12/loesungen/lotto>

Last update: **2026/01/15 07:35**

