

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЕТ
к лабораторной работе 1
по дисциплине «Проектирование программного обеспечения
интеллектуальных систем»

Выполнил:
Проверил:

И.Д.Кондратьева
С. В. Бутрин

Минск 2023

Цель: изучить основные возможности языка Python для разработки программных систем командной строки (CLI)

Вариант: модель банкомата

Для создания интерфейса была использована библиотека click. Пользователь может вводить следующие команды:

- `--number` - необходима для активизации карты.
- `--pin` - необходима для активизации карты.
- `--balance -r` - проверка баланса карты.
- `--rmoney` - выдача наличных (номинал банкноты).
- `--count` - количество купюр.
- `--phone` - зачислить деньги на номер телефона.
- `--new -a` - изменить номер и пинкод карты.
- `--atm -s` - проверить количество и номинал купюр в банкомате.

```
@click.command()
@click.option('--number', help="Number of card")
@click.option('--pin', help="pin of card")
@click.option('--balance')
@click.option('--rmoney')
@click.option('--count', default=1)
@click.option('--phone')
@click.option('--new')
@click.option('--atm')
def beginATM(number, pin, balance, phone, rmoney, count, new, atm):
    assistant = [-1]
    Atm = ATM.ATM(number)
    usersData = Users.Users()

    if number and pin:
        if checkEffectiveCard(number, pin, usersData, assistant):
            if balance == "-r":
                print("Баланс:", str(checkBalance(usersData, assistant)) + " Rub")
            if phone:
                phoneMoney(usersData, assistant, phone)
            if rmoney and count:
                removalMoney(Atm, usersData, assistant, rmoney, count)
            if new == "-a":
                changeCardNumber(usersData, assistant[0] - 1)
        else:
            print("Недостаточно данных!")

    if atm == "-s":
        atmCheck(Atm)
```

Информация о номере карты, пин-коде, номере телефона, а также балансе хранится в текстовом формате "JSON". count - количество карт,

cardNumber - номер карты, Pin - пин-код, cash - баланс карты, phoneNumber - номер телефона привязанный к карте. При изменениях номера и пин-кода карты, снятии денег или пополнение баланса телефона информация в файле обновляется.

```
{ "usersInform": { "count": 5, "items": [
  { "cardNumber": "****9303", "Pin": "2706", "cash": "3344", "phoneNumber": "+375291401013"},
  { "cardNumber": "****5413", "Pin": "5064", "cash": "2756", "phoneNumber": "+375334534294"},
  { "cardNumber": "****8323", "Pin": "6264", "cash": "254", "phoneNumber": "+375291181013"},
  { "cardNumber": "****6151", "Pin": "3554", "cash": "970", "phoneNumber": "+375298463521"},
  { "cardNumber": "****1918", "Pin": "5924", "cash": "876", "phoneNumber": "+375299463434"} ] }
```

Информация о количестве купюр в банкомате и их номинала, также хранится в текстовом формате "JSON"..

```
{ "ATMinfm": {
  "5Rub": ["118", 5],
  "10Rub": ["93", 10],
  "20Rub": ["41", 20],
  "50Rub": ["15", 50],
  "100Rub": ["97", 100],
  "200Rub": ["98", 100],
  "500Rub": ["98", 100]} }
```

Класс АТМ, который инициализируется, считывая информацию из файла. Среди методов класса есть методы getBanknoteNumber (получить количество купюр), getBanknoteValue (получить номинал банкноты), banknoteReduce (уменьшите количество банкнот) и banknoteShow (демонстрация банкнот).

```

class ATM:
    def __init__(self, number):
        jsonfile = open("ATM.json", "r")
        ATMinfm = j.load(jsonfile)
        self.__banknote_5Rub = ATMinfm["ATMinfm"]["5Rub"]
        self.__banknote_10Rub = ATMinfm["ATMinfm"]["10Rub"]
        self.__banknote_20Rub = ATMinfm["ATMinfm"]["20Rub"]
        self.__banknote_50Rub = ATMinfm["ATMinfm"]["50Rub"]
        self.__banknote_100Rub = ATMinfm["ATMinfm"]["100Rub"]
        self.__banknote_200Rub = ATMinfm["ATMinfm"]["200Rub"]
        self.__banknote_500Rub = ATMinfm["ATMinfm"]["500Rub"]
        jsonfile.close()

    def __getBanknoteNumber(self, i):
        if i == 1:
            return self.__banknote_5Rub[0]
        elif i == 2:
            return self.__banknote_10Rub[0]
        elif i == 3:
            return self.__banknote_20Rub[0]
        elif i == 4:
            return self.__banknote_50Rub[0]
        elif i == 5:
            return self.__banknote_100Rub[0]
        elif i == 6:
            return self.__banknote_200Rub[0]
        elif i == 7:
            return self.__banknote_500Rub[0]

    def __getBanknoteValue(self, i):
        if i == 1:
            return self.__banknote_5Rub[1]
        elif i == 2:
            return self.__banknote_10Rub[1]
        elif i == 3:
            return self.__banknote_20Rub[1]
        elif i == 4:
            return self.__banknote_50Rub[1]
        elif i == 5:
            return self.__banknote_100Rub[1]
        elif i == 6:
            return self.__banknote_200Rub[1]
        elif i == 7:
            return self.__banknote_500Rub[1]

```

```

def __banknoteReduce(self, i, number):
    if i == 1:
        self.__banknote_5Rub[0] = str(int(self.__banknote_5Rub[0]) - number)
    elif i == 2:
        self.__banknote_10Rub[0] = str(int(self.__banknote_10Rub[0]) - number)
    elif i == 3:
        self.__banknote_20Rub[0] = str(int(self.__banknote_20Rub[0]) - number)
    elif i == 4:
        self.__banknote_50Rub[0] = str(int(self.__banknote_50Rub[0]) - number)
    elif i == 5:
        self.__banknote_100Rub[0] = str(int(self.__banknote_100Rub[0]) - number)
    elif i == 6:
        self.__banknote_200Rub[0] = str(int(self.__banknote_200Rub[0]) - number)
    elif i == 7:
        self.__banknote_500Rub[0] = str(int(self.__banknote_500Rub[0]) - number)

def __banknoteShow(self):
    print("[5 Rub]:", self.__banknote_5Rub[0], "шт.\n",
          "[10 Rub]:", self.__banknote_10Rub[0], "шт.\n",
          "[20 Rub]:", self.__banknote_20Rub[0], "шт.\n",
          "[50 Rub]:", self.__banknote_50Rub[0], "шт.\n",
          "[100 Rub]:", self.__banknote_100Rub[0], "шт.\n",
          "[200 Rub]:", self.__banknote_200Rub[0], "шт.\n",
          "[500 Rub]:", self.__banknote_500Rub[0], "шт.")

def banknoteReduc(atm, i, number):
    atm.__banknoteReduce(i, number)

def getBanknoteNumb(atm, i):
    return int(atm.__getBanknoteNumber(i))

def getBanknoteVal(atm, i):
    return atm.__getBanknoteValue(i)

def banknoteShow(atm):
    atm.__banknoteShow()

```

Класс Users описывает пользователей карт. В данном классе содержатся методы, позволяющие получить информацию о количестве пользователей, номере карты, пин-коде, балансе, а также номере телефона пользователя. Методы позволяющие изменить номер и пин-код карты.

```

import json as j
import random

class Users:
    def __init__(self):
        jsonfile = open("information.json", "r")
        usersInform = j.load(jsonfile)
        self.__count = usersInform["usersInform"]["count"]
        self.__users = []

        for i in usersInform["usersInform"]["items"]:
            self.__users.append(i)

        jsonfile.close()

    def _getUsersCount(self):
        return self.__count

    def _getUsersCardNumber(self, i):
        return self.__users[i]["cardNumber"]

    def _getCardPin(self, i):
        return self.__users[i]["Pin"]

    def _getBalance(self, i):
        return self.__users[i]["cash"]

    def reduceCash(self, i, reduce):
        self.__users[i]["cash"] = str(int(self.__users[i]["cash"]) - reduce)

    def _getPhoneNumber(self, i):
        return self.__users[i]["phoneNumber"]

    def changeCard(self, i):
        newNumber = [random.randint(1111, 9999), random.randint(1111, 9999), random.randint(1111, 9999), random.randint(1111, 9999)]
        newPin = str(random.randint(1111, 9999))

        number = "****" + str(newNumber[3])

        self.__users[i]["cardNumber"] = number
        self.__users[i]["Pin"] = newPin

        print("Новый номер карты: ", end="")
        for i in newNumber:
            print(i, end=" ")

        print("\nНовый PIN", newPin)

    def _getID(self, i):
        return self.__users[i]["ID"]

    def _getUsersInform(self, i, key):
        return self.__users[i][key]

```