****

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

КОЛЛЕДЖ МНОГОУРОВНЕВОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей

специальность 09.02.07 Информационные системы и программирование

Выполнила студентка:

Топчубаева А.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Гусятинер Л. Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Москва, 2023

СОДЕРЖАНИЕ

[1. ПРОЕКТИРОВАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ 1](#_Toc153963470)

[1.1 Общая информация 1](#_Toc153963471)

* + 1. [Описание бизнес-процесса 1](#_Toc153963472)

[1.1.2. Структура базы данных 1](#_Toc153963473)

[1.1.3. Запросы к базе данных: 2](#_Toc153963474)

[1.2. Построение диаграммы вариантов использования 4](#_Toc153963475)

[1.3 Разработка сценария проекта 4](#_Toc153963476)

[1.4 Построение диаграммы классов 5](#_Toc153963477)

[2. РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ 6](#_Toc153963478)

[2.1 Состав проекта 6](#_Toc153963479)

[2.2 Разработка главного модуля 6](#_Toc153963480)

[2.3 Разработка входящих модулей 6](#_Toc153963481)

[2.4 Тестирование и отладка 9](#_Toc153963482)

[2.5 Дневник практики 10](#_Toc153963483)

[Приложение 1. 13](#_Toc153963484)

[Руководство программиста 13](#_Toc153963485)

[Приложение 2. 14](#_Toc153963486)

[Руководство пользователя 14](#_Toc153963487)

# ПРОЕКТИРОВАНИЕ КОНСОЛЬНОГО ПРИЛОЖЕНИЯ

## Общая информация

Проект представляет собой консольное приложение на Python.

### Описание бизнес-процесса

Клиент регистрируется в банке. Клиент предоставляет свои личные данные и документы, которые подтверждают его личность (паспорт, СНИЛС). Банк проверяет эти данные, устанавливает личность клиента и банковский счет, заносит его в систему банка. Банк фиксирует дату и время активности клиента. Клиент начинает выполнять различные операции (перевод денег, оплата счетов). Банк фиксирует тип транзакции, сумму транзакции, дату и время транзакции. После совершения операции клиент выходит из системы. Банк фиксирует последнюю активность клиента.

### 1.1.2. Структура базы данных

Таблица «Клиент»:

* ID\_клиента
* Фамилия
* Имя
* Отчество
* Серия паспорта
* Номер паспорта
* Телефон
* СНИЛС

Таблица «Банковский счет»

* ID\_счета
* Номер счета
* Баланс счета
* Валюта счета
* ID\_клиента

Таблица «Транзакции»

* ID\_транзакции
* Сумма
* Дата и время
* Код\_Тип транзакции

Таблица «Тип транзакции»

* Код
* Наименование

Таблица «Активности клиента»

* Код активности
* Дата и время активности
* Дата и время последнего доступа
* ID\_клиента

### 1.1.3. Запросы к базе данных:

* Получить данные о клиенте и банковском счете
* Запрос на добавление нового клиента
* Получить сумму, дату и время транзакции по клиенту

***Пример входных файлов:***

«Клиенты.txt»

ID  Фамилия  Имя Отчество СНИЛС Серия паспорта Номер паспорта Телефон

1001;Иванов;Иван;20247229226;4520;754748;+79680849088

1002;Макарова;Анна;20246592657;4520;764898;+79684567233

1003;Павлова;Мария;23634519895;4520;748907;+79034569087

1004;Прядко;Даниила;24567812478;4519;908712;+79067894523

1005;Чижиков;Никита;12389765098;4519;569032;+79084536712

1006;Талибова;Камилла;23076834765;4519;754646;+79082349809

1007;Чернова;Арина;20278962340;4518;754645;+79347890202

«Банковский счет.txt»

ID Код\_клиента Номер счета Баланс Валюта

2001;1001;2345 6789 4568 9080;30000.89;Р

2002;1002;7890 5643 2378 0909;50505.04;Р

2003;1003;0980 6478 3427 8678;10320.00;Р

2004;1004;1234 5678 9012 3456;2489.95;Р

2005;1005;9012 3456 7890 9578;7789.88;Р

2006;1006;4683 3494 4873 9509;62000.00;Р

2007;1007;9484 8373 4243 9348;60500.00;Р

2008;1002;8474 3984 9472 5362;527.02;Р

2009;1006;7457 3839 8372 9023;100.45;Р

2010;1007;3847 9390 3736 0928;215.78;Р

2011;1001;8493 8493 0934 8929;57.99;Р

«Транзакции.txt»

ID Сумма Дата Время Код\_клиента

3001;5678.65;18.11.2023;22:12;4002;1002

3002;150.00;1.11.2023;14:58;4001;1006

3003;65.00;5.11.2023;06:40;4003;1006

3004;790.00;12.11.2023;23:56;4004;1007

3005;5000.00;13.11.2023;13:20;4003;1007

3006;250.00;6.11.2023;09:50;4001;1003

3007;850.00;9.11.2023;17:45;4003;1006

3008;1050.00;10.11.2023;08:54;4002;1004

Запросы:

*Запрос 1: Получить ФИО клиента и номер банковского счета*

Входной файл 1. «Клиенты.txt»

Входной файл 2. «Банковский счет.txt»

Выходной файл

**Клиент: Иванов Иван Иванович**

**Номер счета:** 2345 6789 4568 9080

*Запрос 2: Добавление нового клиента*

Входной файл 1. “Клиенты.txt”

Выходной файл Добавлен новый клиент!

*Запрос 3: Получить сумму, дату и время транзакции по клиенту*

Входной файл 1. «Транзакции.txt»

Выходной файл1. Сумма: 1050.00

Дата: 10.11.2023

Время: 08:54

## 1.2. Построение диаграммы вариантов использования

Диаграмма вариантов использования приведена на Рисунке 1.

Изображение выглядит как диаграмма, текст, круг, линия

Автоматически созданное описание

Рисунок 1 - Диаграмма вариантов использования

## 1.3 Разработка сценария проекта

В данном разделе приведен сценарий использования программы пользователем (Рисунок 2).

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок 2 - Сценарий проекта

* 1. Построение диаграммы классов

В данном разделе находится диаграмма классов приложения.(Рисунок 3)

Изображение выглядит как текст, диаграмма, снимок экрана, число

Автоматически созданное описание

Рисунок 3 - Диаграмма классов

# РАЗРАБОТКА КОНСОЛЬНОГО ПРИЛОЖЕНИЯ

* 1. Состав проекта

Проект состоит из 4 классов: Client, Bank\_account, Transactions, Bank

* 1. Разработка главного модуля

Главный модуль состоит из класса Bank. Он выполняется при запуске программы. Его также можно назвать главным меню.

class Bank:  
 *"""Класс для сбора информации о клиентах"""* def \_\_init\_\_(self, bank\_id: int):  
 *""" """* self.bank\_id = bank\_id  
 self.clients = []  
 self.accounts = []  
 self.transactions = []  
  
 def add\_client(self, client: Client):  
 *"""добавляем клиента в список """* self.clients.append(client)  
  
 def add\_account(self, account: Bank\_account):  
 *""" добавляем банковские счета в список"""* self.accounts.append(account)  
  
 def add\_transacton(self, transaction: Transaction):  
 *""" добавляем транзакции в список"""* self.transactions.append(transaction)

def read\_clients(self, clients\_fname):

*""" считываем данные о клиентах из файла"""* def read\_account(self, account\_fname):  
 *"""считываем данные о банковских аккаунтах из файла"""*

def read\_transaction(self, transaction\_fname):  
 *"""считываем данные о транзакциях из файла"""*

def add\_new\_client(self, id\_client):

def get\_all\_clients(self):

def get\_client\_info(self):

def get\_client\_data(self, client\_id):

Листинг 1 - Главный модуль Bank

* 1. Разработка входящих модулей

Всего разработано 3 модуля:

Модуль client.py, Класс Client, bank\_account.py, класс Bank\_account, transactions.py, класс Transactions

Взаимодействие между модулями можно наглядно увидеть по диаграммам классов из раздела 1.5.

Описание модулей:

Класс Client – представляет собой модель клиента. Он содержит следующие атрибуты:

- client\_id (int): идентификатор клиента;

- surname(str): фамилия клиента;

- name(str): имя клиента;

- snils(int): снилс

- pass\_seria(int): серия паспорта

- pass\_number(int): номер паспорта

- phone(str): номер телефона

Класс Bank\_account – представляет собой модель банковского аккаунта. Он содержит следующие атрибуты:

* id\_acc(int): id счета клиента
* client\_id(int): id клиента
* accNum(str): номер счета
* balance(float): баланс
* currency(str): валюта

Класс Transaction – представляет собой модель транзакции. Он содержит следующие атрибуты:

- id\_trans(int): id транзакции

- summa(float): сумма транзакции

- date(str): дата

- time(str): время

- id\_type(int): тип транзакции

- id\_client(int): id клиента

Рассмотрим функцию для добавления нового клиента в систему:

def add\_new\_client(self, id\_client):  
 print('Введите Фамилию клиента:')  
 surname = input()  
 print('Введите Имя клиента:')  
 name = input()  
 print('Введите СНИЛС клиента')  
 snils = input()  
 print('Введите серию:')  
 pass\_seria = int(input())  
 print('Номер паспорта:')  
 pass\_number = int(input())  
 print('Введите номер телефона клиента:')  
 phone = input()  
 client = Client(id\_client, surname, name, snils, pass\_seria, pass\_number, phone)  
 bank.add\_client(client)  
 client\_f = open('client.txt', 'a', encoding='utf-8')  
 client\_f.write(  
 f"\n {str(id\_client)};{str(surname)};{str(name)};{str(snils)};{str(pass\_seria)};{str(pass\_number)};{str(phone)}\n")  
 client\_f.close()  
 self.clients.append(Client)  
 print('Данные записаны!')  
 self.menu()

Листинг 2. Функция для добавления нового клиента в систему

## Тестирование и отладка

В ходе написания проекта при попытке запустить программу были получены ошибки (Рисунок 5):

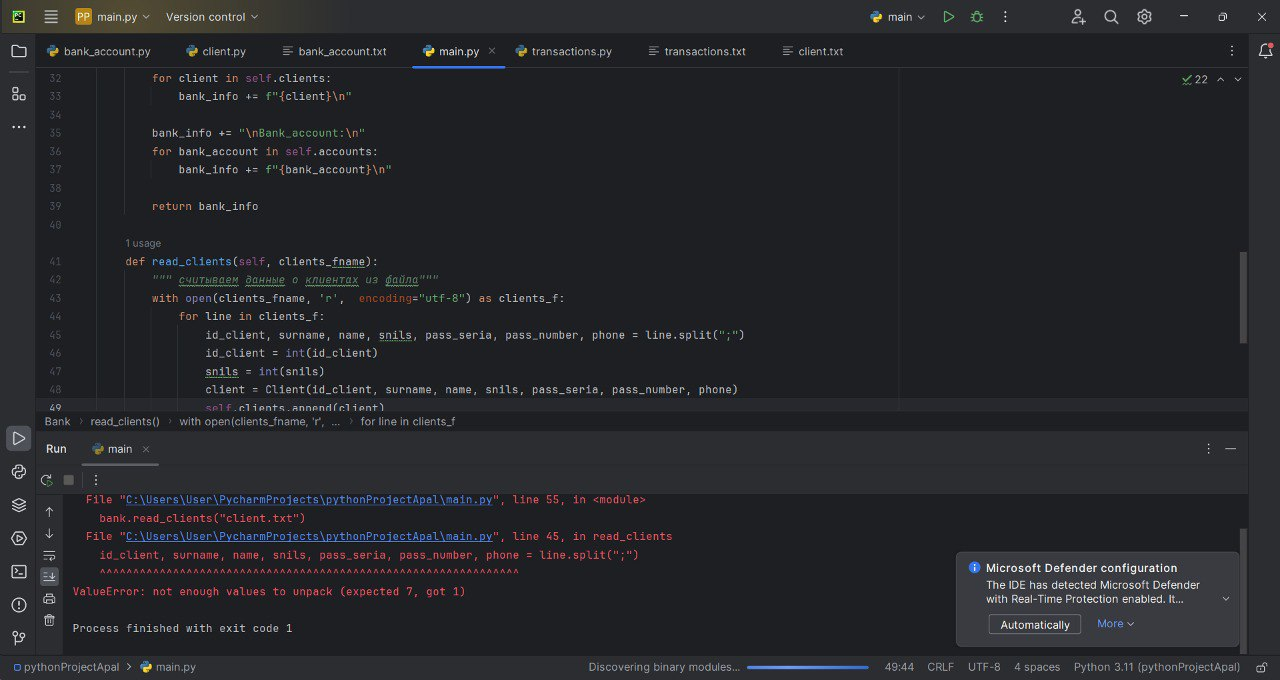


Рисунок 5. Ошибки

При проверке кода были исправлены найденные ошибки, в результате при запуске программы ошибок не было.

## Дневник практики

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 04.09 | Вводный инструктаж |  |
| 04.09 | Формирование алгоритма решения вычислительной задачи |  |
| 11.09 | Формирование алгоритма решения задачи по обработке данных |  |
| 11.09 | Изучение средств визуализации алгоритмов |  |
| 18.09 | Оформление математического алгоритма при помощи программных средств |  |
| 18.09 | Разработка простой спецификации учебной задачи: Введение. Общее описание |  |
| 18.09 | Разработка простой спецификации учебной задачи: Функции системы |  |
| 25.09 | Разработка простой спецификации учебной задачи: Требования к данным |  |
| 25.09 | Разработка простой спецификации учебной задачи: Требования к внешним интерфейсам |  |
| 25.09 | Разработка простой спецификации учебной задачи: Атрибуты качества |  |
| 05.10 | Оформление спецификации программного продукта. Оформление введения и раздела "Общее описание" |  |
| 05.10 | Оформление раздела "Функции системы" |  |
| 12.10 | Оформление раздела "Требования к внешним данным" |  |
| 12.10 | Оформление раздела "Требования к внешним интерфейсам" |  |
| 19.10 | Разработка классов главного модуля |  |
| 19.10 | Разработка методов классов и функций главного модуля |  |
| 26.10 | Разработка классов подсистемы вычислений |  |
| 26.10 | Разработка классов подсистемы связи с внешними данными |  |
| 26.10 | Разработка классов подсистемы отчётов |  |
| 02.11 | Провести тестирование главного модуля |  |
| 09.11 | Провести тестирование модулей |  |
| 09.11 | Провести тестирование программного продукта |  |
| 09.11 | Разработать руководство программиста: Назначение и условия применения программ |  |
| 23.11 | Разработать руководство программиста: Характеристика программы — |  |
| 23.11 | Разработать руководство программиста: Обращение к программе — |  |
| 23.11 | Разработать руководство программиста: Входные и выходные данные |  |
| 23.11 | Разработать руководство программиста: Сообщения |  |
| 30.11 | Разработать руководство пользователя: Введение. Назначение и условия применения. |  |
| 30.11 | Разработать руководство пользователя: Подготовка к работе. |  |
| 07.12 | Разработать руководство пользователя: Описание операций. |  |
| 07.12 | Разработать руководство пользователя: Аварийные ситуации. |  |
| 07.12 | Разработать руководство пользователя: Рекомендации по освоению. |  |
| 20.12 | Промежуточная аттестация: дифференцированный зачет |  |