****

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 студент:

Талибова К. Т.

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

Гусятинер Л. Б.

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

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

Москва, 2023

**Содержание**

[Раздел 1. Проектирование консольного приложения 1](#_Toc153957438)

[1.1 Общая информация: 1](#_Toc153957439)

[1.2 Изучение входной и выходной документации 3](#_Toc153957440)

[1.3 Разработка требований к проекту. 4](#_Toc153957441)

[Построение диаграммы использования. 4](#_Toc153957442)

[1.4 Разработка сценария проекта 6](#_Toc153957443)

[Раздел 2. Разработка консольного приложения 7](#_Toc153957444)

[2.1 Разработка главного модуля 7](#_Toc153957445)

[2.2 Разработка входящих модулей 10](#_Toc153957446)

[2.3 Тестирование и отладка 11](#_Toc153957447)

[2.4 Дневник 12](#_Toc153957448)

[Использованные источники 15](#_Toc153957449)

[**Приложение 1. Руководство программиста** 16](#_Toc153957450)

[**Приложение 2. Руководство пользователя** 17](#_Toc153957451)

[**Приложение 3. Листинги** 18](#_Toc153957452)

Раздел 1. Проектирование консольного приложения

## 1.1 Общая информация:

Клиент приходит в фитнес клуб и обращается к сотруднику на ресепшене. Администратор предлагает клиенту выбрать тип абонемента и длительность. Клиент выбирает подходящий для себя вариант и заполняет анкету с персональными данными. Сотрудник клуба заполняет информацию о покупке в базе данных. Клиент производит оплату выбранным способом либо картой, либо наличными. Администартор клуба подтверждает оплату и выдает клиенту бумажный билет. В день занятий клиент приходит в фитнес-клуб в выбранное время и предъявляет бумажный билет на регистрации в клубе. Сотрудник клуба проверяет билет и регистрирует клиента на занятие. Клиент начинает занятия по выбранному абонементу.

Структура базы данных

1. Таблица "Клиент":

* Идентификатор
* Фамилия
* Имя
* Дата рождения
* Пол
* Логин
* Пароль
* Номер телефона

1. Таблица "Клубная карта":

* Идентификатор
* Название
* Тип
* Имя клиента
* Дата рождения
* Пол клиента
* Стоимость

1. Таблица "Платежи"

* Идентификатор
* Номер телефона клиента
* Фамилия и имя клиента
* Email
* Пол
* Дата рождения
* Есть ли опыт в занятии фитнес-клуба
* Оплата (частями, рассрочку, полностью)
* Номер карты

Запросы к базе данных:

1. Фамилия и имя клиента и тип его Клубной карты.

2. Фамилия и имя клиента и данные о том, есть ли опыт в занятии фитнес клубе.

## 1.2 Изучение входной и выходной документации

Запрос: ФИО клиента и тип его Клубной карты

Входной файл 1

⋅ «Клиенты»

ID Фамилия Имя Дата рождения Пол Логин Пароль Номер телефона

01; Павлова Мария; 20.11.2004; Ж; maria; 8808; 89945558887

02; Иванов Александр; 22.12.2003; М; pasha; 7707; 89285558887

04; Чернова Арина; 12.01.2001; Ж; arina; 6606; 89256667779

Входной файл 2

⋅ «Клубная карта»

ID Название Тип Имя клиента Дата рождения Пол клиента Стоимость

03; Light; Подарочная; Мария; 20.11.2004; Ж; 3500

06; Smart; Бонусная; Александр; 22.12.2003; M; 1500

05; Smart; Подарочная; Арина; 12.12.2001; Ж; 3500

Выходной файл

**Фамилия и имя клиента:**

01. Павлова Мария

**Тип клубной карты:**

03. Подарочная

Запрос: Фамилия и имя клиента и данные о том, есть ли опыт в занятии фитнес клубе.

Входной файл 1

⋅ «Клиенты»

ID Фамилия Имя Дата рождения Пол Логин Пароль Номер телефона

01; Павлова Мария; 20.11.2004; Ж; maria; 8808; 89945558887

02; Иванов Александр; 22.12.2003; М; pasha; 7707; 89285558887

04; Чернова Арина; 12.01.2001; Ж; arina; 6606; 89256667779

Входной файл 2

⋅ «Платежи»

01; 89945558887; Павлова Мария; pavlova@inbox.ru; Ж; 20.11.2004; нет опыта; полностью; 2024 5678 9800 8346

02; 89285558887; Иванов Александр; ivanov@inbox.ru; М; 22.12.2003; нет опыта; полностью; 2874 8957 9345 2575

04; 89256667779; Чернова Арина; cheri@mail.ru; 12.12.2001; есть опыт; частями; 2023 5689 9300 2545

Выходной файл

**Талибова Камилла Талибовна;** Нет опыта

**Иванов Александр Иванович**; Нет опыта

## 1.3 Разработка требований к проекту.

## Построение диаграммы использования.

* В таблицах "Менеджеры", "Клиенты" и "Заказы" не должно быть двух записей с одинаковым идентификатором.
* Поля в таблицах должны корректно ссылаться на другие.
* Дата заказа должна быть в определенном диапазоне.

Диаграмма использования приведена на Рисунке 1.

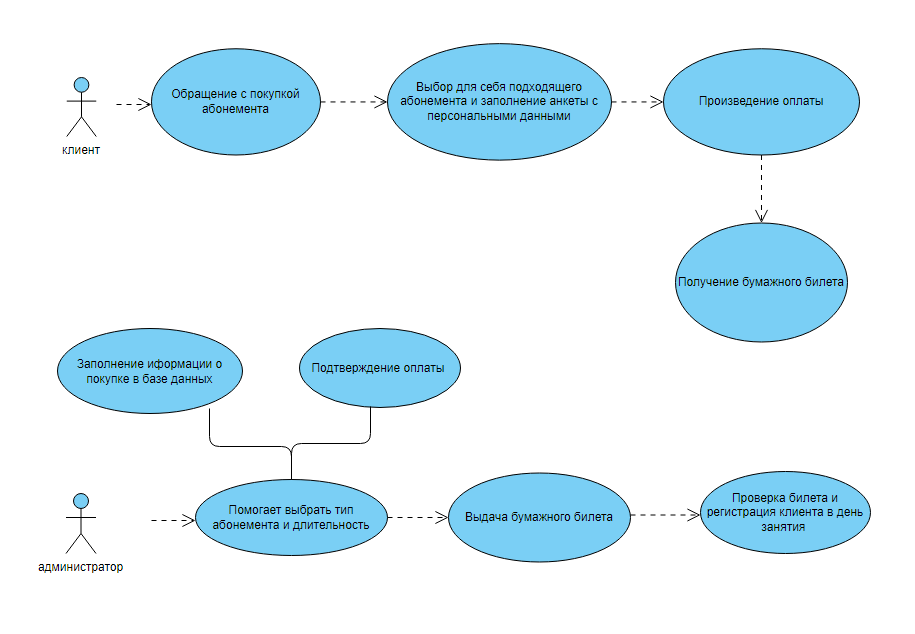


Рисунок 1 - Диаграмма использования

## 1.4 Разработка сценария проекта

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

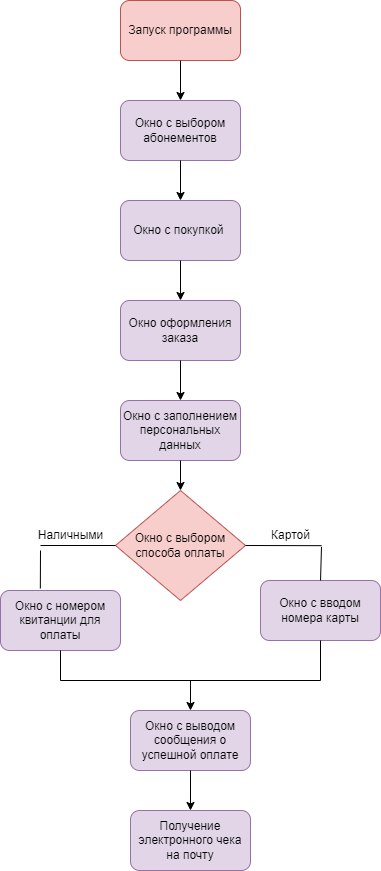


Рисунок 2 - Сценарий проекта

Раздел 2. Разработка консольного приложения

## 2.1 Разработка главного модуля

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

Листинг 1. Главный модуль Main

from client import Client  
from club\_card import Club\_card  
from payments import Payments  
  
  
class Fitness:  
  
  
 def \_\_init\_\_(self, fitness\_id: int):  
 *""" """* self.fitness\_id = fitness\_id  
 self.clients = []  
 self.club\_card = []  
 self.payments = []  
  
 def add\_client(self, client: Client):  
 *"""добавляем клиента в список """* self.clients.append(client)  
  
 def add\_club\_card(self, club\_card: Club\_card):  
 *"""добавляем клубную карту в список """* self.club\_card.append(club\_card)  
  
 def add\_payments(self, payments: Payments):  
 *""" добавляем платежи в спсиок"""* self.payments.append(payments)  
  
 def \_\_str\_\_(self):  
 fitness\_info = f"Fitness ID: {self.fitness\_id}\n\n"  
  
 fitness\_info += "Club\_card:\n"  
 for club\_card in self.club\_cards:  
 fitness\_info += f"{club\_card}\n"  
  
 fitness\_info += "Clients:\n"  
 for client in self.clients:  
 fitness\_info += f"{client}\n"  
  
 fitness\_info += "Payments:\n"  
 for payment in self.payments:  
 fitness\_info += f"{payment}\n"  
  
 return f"{self.clients} {self.payments} {self.club\_cards}"  
  
  
  
 def read\_club\_cards(self, club\_cards\_fname):  
 with open(club\_cards\_fname, "r", encoding="utf-8") as club\_cards\_f:  
 for line in club\_cards\_f:  
 values = line.split(";")  
 print(line)  
 if len(values) == 7:  
 club\_card\_id, name, card\_type, client\_name, gender, data, price = line.split(";")  
 club\_card\_id = int(club\_card\_id)  
 name = str(name)  
 card\_type = str(card\_type)  
 client\_name = str(client\_name)  
 gender = str(gender)  
 data = str(data)  
 price = int(price)  
  
 club\_card = Club\_card(club\_card\_id, name, card\_type, client\_name, gender, data, price)  
 self.club\_card.append(club\_card)  
  
 def read\_clients(self, clients\_fname):  
 with open(clients\_fname, 'r', encoding="utf-8") as clients\_f:  
 for line in clients\_f:  
 values = line.split(";")  
 print(line)  
 if len(values) == 7:  
 client\_id, name, data, gender, login, password, number\_phone = line.split(";")  
 client\_id = int(client\_id)  
 name = str(name)  
 data = str(data)  
 gender = str(gender)  
 login = str(login)  
 password = int(password)  
 number\_phone = int(number\_phone)  
 client = Client(client\_id, name, data, gender, login, password, number\_phone)  
 self.clients.append(client)  
  
 def read\_payments(self, payments\_fname):  
 with open(payments\_fname, 'r', encoding="utf-8") as payments\_f:  
 for line in payments\_f:  
 values = line.split(";")  
 print(line)  
 if len(values) == 9:  
 payments\_id, number\_phone\_client, full\_name\_client, email, gender, data, experience\_lesson, payment,card\_number = line.split(";")  
 payments\_id = int(payments\_id)  
 number\_phone\_client = int(number\_phone\_client)  
 full\_name\_client = str(full\_name\_client)  
 email = str(email)  
 gender = str(gender)  
 data = int(data)  
 experience\_lesson = str(experience\_lesson)  
 payment = str(payment)  
 card\_number = int(card\_number)  
 payment = Payments(payments\_id, number\_phone\_client, full\_name\_client, email, gender, data, experience\_lesson, payment,card\_number)  
 self.payments.append(payment)  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 fitness = Fitness(1)  
 fitness.read\_clients("client.txt")  
 fitness.read\_club\_cards("club\_card.txt")  
 for client in fitness.clients:  
 for club\_card in fitness.club\_card:  
 if club\_card.client\_name == client.name:  
 print(f"ФИО клиента и тип клубной карты:{client.name} - {club\_card.card\_type}\n")  
  
 fitness = Fitness(1)  
 fitness.read\_payments("payments.txt")  
 fitness.read\_clients("client.txt")  
 for client in fitness.clients:  
 for payment in fitness.payments:  
 if client == payment.client and payment.experience\_lesson:  
 print(f"{payment.full\_name\_client}-есть опыт")  
 elif client == payment.client and not payment.experience\_lesson:  
 print(f"{payment.full\_name\_client}-нет опыта")

## Разработка входящих модулей

Всего разработано 3 модуля:

* Класс Client
* Класс Club\_card
* Класс Payments

Описание модулей:

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

* client\_id (int): идентификатор клиента.
* name (str): имя клиента.
* data (data): дата рождения
* gender (str): пол
* login (str): логин
* password(str): пароль
* number\_phone(int): номер телефона

Класс Club\_card - представляет модель клубной карты. Он имеет следующие атрибуты:

* club\_card\_id (int): идентификатор клубной карты
* name (str): имя клубной карты
* card\_type (str): тип клубной карты
* client\_name (str): имя клиента
* data (data): дата рождения
* gender (str): пол
* price (int): цена

Класс Payments моделирует платежи. Он включает в себя следующие атрибуты:

* payments\_id(str): идентификатор платежей
* number\_phone\_client (int): номер телефона клиента
* full\_name\_client (str): ФИО клиента
* email (str): почта
* gender (str): пол
* data (data): дата рождения
* experience\_lesson (str): есть ли опыт в занятии фитнес-клуба
* payment (str): платежи
* card\_number (int): номер карты

## 2.3 Тестирование и отладка

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

Рисунок 5. Ошибки

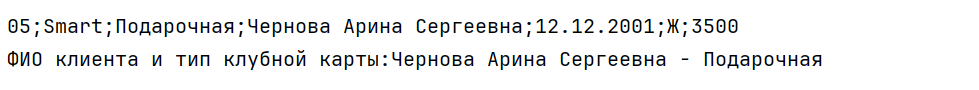
При проверке кода были исправлены найденные ошибки, в результате при запуске программы ошибок не было (Рисунок 6):

Рисунок 6. Успешная сборка

## 2.4 Дневник

|  |  |  |
| --- | --- | --- |
| 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. |  |
| 14.12 | Промежуточная аттестация: дифференцированный зачет |  |

## Использованные источники

1. Сайт языка Python  
   [www.xxx.yyy](http://www.xxx.yyy)
2. …

**Приложение 1. Руководство программиста**

**Приложение 2. Руководство пользователя**

**Приложение 3. Листинги**