****

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

Выполнил студент:

Алёхин Д.А.

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

Гусятинер Л. Б.

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

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

Москва, 2023

**Содержание**

[**Раздел 1. Проектирование консольного приложения** 1](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971401)

[1.1 Общая информация 1](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971402)

[1.2 Изучение входной и выходной документации 3](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971403)

[1.3 Разработка требований к проекту. 5](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971404)

[**Раздел 2. Разработка консольного приложения** 8](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971407)

[2.1 Разработка главного модуля 8](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971408)

[2.2 Разработка входящих модулей 10](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971409)

[2.3 Тестирование и отладка 13](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971410)

[2.4 Дневник 14](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971411)

[**Использованные источники** 16](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971412)

[**Приложение 1. Листинг класса Order** 17](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971413)

[**Приложение 2. Листинг класса End Invoice** 17](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971414)

[**Приложение 3. Листинг класса Delivery** 17](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971415)

[**Приложение 4. Листинг класса Check** 18](file:///C:\Users\student\Downloads\уп%20алехин.docx#_Toc153971416)

**Раздел 1. Проектирование консольного приложения**

1.1 Общая информация

Проект представляет собой консольное приложение на языке Python.

***Описание бизнес-процесса***

«Процесс оформления Доставки»

1.Клиент пришел в магазин, выбрал товар и выбирает способ доставки и указывает адрес.

2. Далее товар передается курьеру, и он доставляет товар в указанный адрес.

3. После доставки клиент может отказаться от товара, если он поврежден, или принять товар.

4. Далее клиент оплачивает доставку товара, в случае принятия заказа, получает чек.

***Структура базы данных***

Таблица 1

Заказ

-id заказа

-id доставки

-товар

-фио

-цена

Таблица 2

Доставка

-id доставки

-имя курьера

-адрес

Таблица 3

Чек

-id чек

- id заказа

-имя курьера

-адрес

- товар

-цена

-ФИО

**Запросы к базе данных**

1. Получить данные о заказе

В данном проекте я работал с файлами и консолью, автоматизировал процесс документации.

## 1.2 Изучение входной и выходной документации

*Запрос: Получить данные о заказе*

Входной файл 1

«Заказ»

id заказа id доставки товар ФИО цена

1; 1; самокат; Таврин Валерий Иванович; 8990

2; 2; велосипед; Тихонов Иван Андреевич; 17500

Входной файл 2

«доставка»

id доставки имя курьера адрес

1; Виталий; ул пушкина

2; Родион; ул колотушкина

Выходной файл 1

«чек»

Id чек id заказа имя курьера адрес товар цена ФИО

1 1 Виталий ул пушкина самокат Таврин 8990 Валерий Иванович

2 2 Родион ул колотушкина велосипед 17500 Тихонов Иван Андреевич

## 1.3 Разработка требований к проекту.

Построение диаграммы использования.

В таблицах «Заказ», «Доставка», «Чек» не должно быть двух записей с одинаковым идентификатором.

Поля таблиц должны корректно ссылаться на другие.

**Раздел 2. Разработка консольного приложения**

## 2.1 Разработка главного модуля

Главный модуль состоит из класса Order. Он выполняется при запуске программы

Листинг 1. Главный модуль Order

class Order:

from delivery import Delivery

from receipt import Receipt

class Order:

def \_\_init\_\_(self, order\_id, delivery\_id, product, full\_name, price):

self.order\_id = order\_id

self.delivery\_id = delivery\_id

self.product = product

self.full\_name = full\_name

self.price = price

def create\_receipt(order, delivery):

return Receipt(

receipt\_id=order.order\_id,

order\_id=order.order\_id,

courier\_name=delivery.courier\_name,

address=delivery.address,

product=order.product,

price=order.price,

full\_name=order.full\_name

)

# функция считывания заказа

def read\_orders(file\_path):

orders = []

with open(file\_path, 'r', encoding='utf-8') as file:

lines = file.readlines()[1:] # skip the header

for line in lines:

data = line.strip().split('; ')

order = Order(\*data)

orders.append(order)

return orders

# функция считывания доставки

def read\_delivery(file\_path):

deliveries = []

with open(file\_path, 'r', encoding='utf-8') as file:

lines = file.readlines()[1:]

for line in lines:

data = line.strip().split('; ')

delivery = Delivery(\*data)

deliveries.append(delivery)

return deliveries

# функция создания чека

def create\_receipts(orders, deliveries):

receipts = []

for order in orders:

corresponding\_delivery = next(delivery for delivery in deliveries if delivery.delivery\_id == order.delivery\_id)

receipt = create\_receipt(order, corresponding\_delivery)

receipts.append(receipt)

return receipts

# функция записи чека

def write\_receipts(file\_path, receipts):

with open(file\_path, 'w', encoding='utf-8') as file:

file.write("Id чек id заказа имя курьера адрес товар цена ФИО\n")

for receipt in receipts:

file.write(f"{receipt.receipt\_id} {receipt.order\_id} {receipt.courier\_name} {receipt.address} {receipt.product} {receipt.price} {receipt.full\_name}\n")

if \_\_name\_\_ == "\_\_main\_\_":

order\_file\_path = "order.txt"

delivery\_file\_path = "delivery.txt"

receipt\_file\_path = "receipt.txt"

orders = read\_orders('order.txt')

deliveries = read\_delivery('delivery.txt')

receipts = create\_receipts(orders, deliveries)

write\_receipts('receipt.txt', receipts)

## 2.2 Разработка входящих модулей

Всего разработано 3 модуля:

* Класс «Delivery»
* Класс «Order»
* Класс «Receipt»

Описание модулей:

Класс «Order» – представляет собой модель заказа. Он содержит следующие атрибуты:

- order\_id (int): идентификатор.

- delivery\_id (str): идентификатор доставки.

- product (str): название продукта.

- full\_name (str): имя заказчика.

- price (int): цена товара.

Класс «Delivery» – представляет модель доставки. Он имеет следующие атрибуты:

- delivery\_id (int): идентификатор доставки.

- courier\_name (str): имя курьера.

- address (str): адрес заказчика.

Класс «Receipt» – моделирует чек. Он включает в себя следующие атрибуты:

- receipt\_id (int): идентификатор.

- order\_id (int): идентификатор заказа.

- courier\_name (str): имя курьера.

- address (str): адрес заказчика.

- product (str): название продукта.

- price (float): сумма заказа.

- full\_name (str): полное имя заказчика.

Рассмотрим функцию для получения информации о менеджере клиента на основе идентификатора клиента и её блок-схему:

Листинг 2. Функция для получения информации о заказе:

def read\_orders(file\_path):

orders = []

with open(file\_path, 'r', encoding='utf-8') as file:

lines = file.readlines()[1:] # skip the header

for line in lines:

data = line.strip().split('; ')

order = Order(\*data)

orders.append(order)

return orders

Листинг 3. Функция считывания доставки:

def read\_delivery(file\_path):

deliveries = []

with open(file\_path, 'r', encoding='utf-8') as file:

lines = file.readlines()[1:]

for line in lines:

data = line.strip().split('; ')

delivery = Delivery(\*data)

deliveries.append(delivery)

return deliveries

Листинг 4. Функция создания чека:

def create\_receipts(orders, deliveries):

receipts = []

for order in orders:

corresponding\_delivery = next(delivery for delivery in deliveries if delivery.delivery\_id == order.delivery\_id)

receipt = create\_receipt(order, corresponding\_delivery)

receipts.append(receipt)

return receipts

Листинг 5. Функция записи чека:

def write\_receipts(file\_path, receipts):

with open(file\_path, 'w', encoding='utf-8') as file:

file.write("Id чек id заказа имя курьера адрес товар цена ФИО\n")

for receipt in receipts:

file.write(f"{receipt.receipt\_id} {receipt.order\_id} {receipt.courier\_name} {receipt.address} {receipt.product} {receipt.price} {receipt.full\_name}\n")

## 2.3 Тестирование и отладка

В ходе написания проекта были протестированы входные данные

Входной файл 1 «Заказ»

id заказа id доставки товар ФИО цена

1; 1; самокат; Таврин Валерий Иванович; 8990

2; 2; велосипед; Тихонов Иван Андреевич; 17500

Входной файл 2 «Доставка»

id доставки имя курьера адрес

1; Виталий; ул пушкина

2; Родион; ул колотушкина

Выходной файл 1 «чек»

Id чек id заказа имя курьера адрес товар цена ФИО

1 1 Виталий ул пушкина самокат Таврин 8990 Валерий Иванович

2 2 Родион ул колотушкина велосипед 17500 Тихонов Иван Андреевич

При проверке кода были исправлены найденные ошибки, в результате при запуске программы ошибок не было (Рисунок 1):

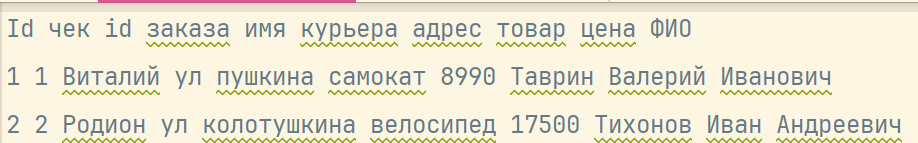


Рисунок 1. Успешная сборка

## 2.4 Дневник

|  |  |  |
| --- | --- | --- |
| Дата | Содержание работ | Отметка о выполнении |
| 01.09 | Вводный инструктаж. Формирование алгоритма решения вычислительной задачи. Формирование алгоритма решения задачи по обработке данных |  |
| 04.09 | Изучение средств визуализации алгоритмов |  |
| 09.09 | Оформление математического алгоритма при помощи программных средств |  |
| 11.09 | Разработка простой спецификации учебной задачи: Введение. Общее описание |  |
| 16.09 | Разработка простой спецификации учебной задачи: Функции системы |  |
| 22.09 | Разработка простой спецификации учебной задачи: Требования к данным. Требования к внешним интерфейсам. Атрибуты качества |  |
| 29.09 | Оформление спецификации программного продукта. Оформление введения и раздела «Общее описание». Оформление раздела «Функции системы». Оформление раздела «Требования к внешним данным» |  |
| 04.10 | Оформление раздела «Требования к внешним интерфейсам». Оформление раздела «Атрибуты качества» |  |
| 11.10 | Разработка классов главного модуля. Разработка методов классов и функций главного модуля |  |
| 18.10 | Разработка классов подсистемы вычислений. Разработка классов подсистемы связи с внешними данными |  |
| 25.10 | Разработка классов подсистемы отчетов. Разработать тестовые наборы |  |
| 01.11 | Провести тестирование главного модуля. Провести тестирование модулей |  |
| 08.11 | Провести тестирование программного продукта. По результатам тестирования провести рефакторинг программного продукта |  |
| 20.12 | Промежуточная аттестация: дифференцированный зачет |  |

## Использованные источники

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

<https://www.jetbrains.com/ru-ru/pycharm/>

1. Моделирование бизнес-процессов

https://www.businessstudio.ru/products/business\_studio/notations/