****

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

Выполнили студенты:

Рябова Е.Д.

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

Гусятинер Л. Б.

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

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

Москва, 2023

**Содержание**

[Раздел 1. Проектирование консольного приложения 1](#_Toc153358240)

[1.1 Техническое задание 1](#_Toc153358241)

[1.1.1. Описание бизнес-процесса 1](#_Toc153358242)

[1.1.2. Структура базы данных 1](#_Toc153358243)

[1.1.3. Запросы к базе данных: 4](#_Toc153358244)

[Раздел 2. Разработка консольного приложения 5](#_Toc153358245)

[2.1. Состав проекта 5](#_Toc153358246)

[2.2 Разработка главного модуля 5](#_Toc153358247)

[2.3 Разработка входящих модулей 9](#_Toc153358248)

[2.4 Тестирование и отладка 9](#_Toc153358249)

[2.5 Дневник 11](#_Toc153358250)

[Использованные источники 12](#_Toc153358251)

[Приложение 1. Руководство программиста 13](#_Toc153358252)

[Приложение 2. Руководство пользователя 14](#_Toc153358253)

[Приложение 3. Листинги 15](#_Toc153358254)

# Раздел 1. Проектирование консольного приложения

## Техническое задание

### Описание бизнес-процесса

### Процесс учета заявок на поддержку 1С.

### 1. Пользователь отправляет заявку на поддержку 1С, указывая контактные данные и описание проблемы.

### 2. Система получает заявку и записывает ее в базу данных заявок на поддержку.

### 3. Сотрудник поддержки просматривает новые заявки и выбирает одну для обработки.

### 4. Сотрудник поддержки анализирует проблему и предлагает решение.

### 5. Сотрудник поддержки вносит информацию о решении проблемы в заявку и закрывает ее.

### 6. Система отправляет уведомление пользователю о решении проблемы.

### **1.1.2. Структура базы данных**

# **Таблица "Заявка"**

# - ID заявки

# - Контактные данные пользователя

# - Описание проблемы

# - Статус заявки (открыта/закрыта)

# **Таблица "Решение"**

# - ID решения

# - ID заявки

# - Описание решения

# - Дата решения

# **1.1.3 Запросы к базе данных**

# 1. Получение всех открытых заявок на поддержку.

# 2. Получение всех закрытых заявок на поддержку.

# 3. Получение общего количества заявок на поддержку.

# 4. Получение заявок на поддержку, отсортированных по дате создания.

# 5. Получение заявок на поддержку с определенным статусом.

# **1.2 Тестовые наборы**

# Входной файл 1

# Заявка 001

# -001

# -Иванов Иван Иванович

# -ivanov@example.com

# -Проблема с отчетом по продажам

# Входной файл 2

# Заявка 002

# -002

# -Петров Петр Петрович

# -petrov@example.com

# -Проблема с импортом данных из Excel

# Выходной файл 1

# Заявка 001

# -001

# -Иванов Иван Иванович

# -ivanov@example.com

# -Проблема с отчетом по продажам

# -Заявка закрыта

# Выходной файл 2

# Заявка 002

# -002

# -Петров Петр Петрович

# -petrov@example.com

# -Проблема с импортом данных из Excel

# -Решение: необходимо обновить драйвер базы данных.

# 1.3 Запросы к базе данных

# 1. Получение всех открытых заявок на поддержку.

# 2. Получение всех закрытых заявок на поддержку.

# 3. Получение общего количества заявок на поддержку.

# 4. Получение заявок на поддержку, отсортированных по дате создания.

# 5. Получение заявок на поддержку с определенным статусом.

# Раздел 2. Разработка консольного приложения

## 2.1. Состав проекта

1) main.py

2) database.py

## 2.2 Разработка главного модуля

Главный модуль состоит из функции “main”. Она вызывается при запуске программы. Её также можно назвать главным меню.

Листинг 1. Главный модуль main (код)

class Ticket:

def \_\_init\_\_(self, customer, issue, priority):

self.customer = customer

self.issue = issue

self.priority = priority

self.status = "New"

```

support\_system.py:

```python

from ticket import Ticket

class SupportSystem:

def \_\_init\_\_(self):

self.tickets = []

def create\_ticket(self, customer, issue, priority):

ticket = Ticket(customer, issue, priority)

self.tickets.append(ticket)

self.notify\_support\_personnel(ticket)

def notify\_support\_personnel(self, ticket):

print(f"Notification: New ticket created - {ticket.issue}")

```

main.py:

```python

from support\_system import SupportSystem

def main():

support = SupportSystem()

support.create\_ticket("Client1", "Can't run report", "High")

support.create\_ticket("Client2", "Error message on login", "Low")

for ticket in support.tickets:

print(f"Ticket from {ticket.customer}: {ticket.issue}, Priority: {ticket.priority}, Status: {ticket.status}")

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

main()

## 2.3 Разработка входящих модулей

Всего разработано 6 модулей:

Модуль client.py, Класс Ticket

Модуль product.py, Класс SupportSystem

Описание модулей:

Ticket - это класс, представляющий сущность заявки в вашей системе.

- client\_id (int): идентификатор клиента.

SupportSystem - это класс, отвечающий за управление заявками и уведомления персонала поддержки.

## 2.4 Тестирование и отладка

В ходе написания проекта при попытке выбрать функцию были получены ошибки (Рисунок 5):

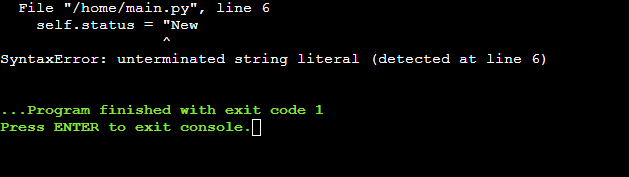


Рисунок 5. Ошибки

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

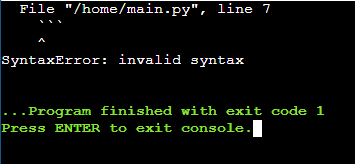


Рисунок 6. Успешная сборка

## 2.5 Дневник

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Заявка | Описание проблемы | Статус заявки |

# 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