

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВВГУ»)

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

РЕКОМЕНДОВАНО

к защите

Руководитель проекта

ст. преподаватель

— М.В. Водяницкий

ОТЧЕТ ПО ПРОЕКТУ

Система управления автозаправочной станцией
ИТ-03-01-123456.2024-с.01.000.ПР

Студент

гр. БИН-25-3

Р.А. Сакс

Руководитель проекта

ст. преподаватель

М.В. Водяницкий

Аннотация

Данный отчет представляет результаты разработки консольной системы управления автозаправочной станцией (АЗС). Система реализована на языке Python и предназначена для автоматизации основных процессов работы АЗС, включая обслуживание клиентов, контроль запасов топлива, управление оборудованием и ведение статистики.

Система обеспечивает работу с различными типами топлива, управление цистернами и заправочными колонками, обработку транзакций, а также включает механизмы безопасности и аварийного отключения. Все данные сохраняются в файловой системе в формате JSON, что обеспечивает сохранение состояния между сеансами работы.

Разработанная система демонстрирует практическое применение принципов объектно-ориентированного программирования, работы с файлами и построения консольных интерфейсов.

Abstract

This report presents the results of developing a console-based automated fuel station management system. The system is implemented in Python and is designed to automate the main processes of a fuel station, including customer service, fuel inventory control, equipment management, and statistics keeping.

The system supports various fuel types, manages storage tanks and fuel dispensers, processes transactions, and includes safety mechanisms and emergency shutdown procedures. All data is stored in the file system in JSON format, ensuring state persistence between sessions.

The developed system demonstrates the practical application of object-oriented programming principles, file handling, and console interface design.

Задание на проект

Разработать консольную систему управления автозаправочной станцией, которая должна обеспечивать следующие функции:

1) Обслуживание клиентов (продажа топлива):

- Выбор заправочной колонки
- Выбор типа топлива
- Расчет стоимости
- Проверка доступности топлива
- Фиксация транзакции

2) Управление цистернами:

- Мониторинг уровня топлива
- Автоматическое отключение при низком уровне
- Ручное включение/отключение
- Пополнение запасов

3) Управление оборудованием:

- Контроль состояния заправочных колонок
- Управление схемой подключения цистерн к колонкам

4) Статистика и отчетность:

- Учет проданного топлива
- Финансовая статистика
- История операций

5) Безопасность:

- Аварийный режим
- Резервное копирование данных
- Восстановление состояния

6) Технические требования:

- Консольный интерфейс
- Сохранение данных в файлы

- Модульная архитектура
- Обработка ошибок

Содержание

Введение	3
1 Анализ предметной области и проектирование системы	4
1.1 Основные сущности и их характеристики	4
1.2 Схема подключения оборудования	4
1.3 Требования к функциональности	5
2 Архитектура и реализация системы	6
2.1 Структура классов	6
2.2 Ключевые алгоритмы	7
2.3 Хранение данных	8
3 Реализация основных функций	9
3.1 Модуль обслуживания клиентов	9
3.2 Модуль управления цистернами	10
3.3 Модуль статистики и отчетности	11
4 Тестирование и результаты	12
4.1 Методология тестирования	12
4.2 Результаты тестирования	12
4.3 Обработка исключительных ситуаций	13
Заключение	14
Список использованных источников	15
Приложение А Исходный код программы	16
Приложение Б Примеры работы системы	17

Введение

Автоматизация процессов управления автозаправочными станциями является важной задачей в современных условиях развития топливно-энергетического комплекса. Эффективное управление запасами топлива, контроль за оборудованием и автоматизация расчетов с клиентами позволяют повысить производительность, снизить операционные издержки и минимизировать человеческие ошибки.

Целью данного проекта является разработка программной системы для управления автозаправочной станцией, которая позволит автоматизировать ключевые бизнес-процессы, обеспечить надежный учет операций и предоставить инструменты для анализа эффективности работы станции.

Основные задачи проекта:

- 1) Анализ предметной области и требований к системе управления АЗС
- 2) Проектирование архитектуры системы и структуры данных
- 3) Разработка классов для представления основных сущностей (цистерны, колонки, транзакции)
- 4) Реализация функционала обслуживания клиентов и управления оборудованием
- 5) Разработка механизмов сохранения и восстановления состояния системы
- 6) Создание консольного пользовательского интерфейса
- 7) Тестирование и отладка системы

Система разработана на языке программирования Python, что обеспечивает кроссплатформенность, простоту поддержки и расширения функциональности.

1 Анализ предметной области и проектирование системы

1.1 Основные сущности и их характеристики

Автозаправочная станция представляет собой комплекс оборудования и бизнес-процессов, которые могут быть формализованы в виде следующих основных сущностей:

- 1) **Топливо** - характеризуется типом (АИ-92, АИ-95, АИ-98, ДТ) и ценой за литр
- 2) **Цистерна** - емкость для хранения топлива определенного типа, имеет максимальный объем, текущий уровень и состояние (включена/отключена)
- 3) **Заправочная колонка** - оборудование для отпуска топлива клиентам, может быть подключена к нескольким цистернам
- 4) **Транзакция** - запись о совершенной операции (продажа, пополнение, перекачка и т.д.)
- 5) **Статистика** - накопленные данные о работе АЗС

1.2 Схема подключения оборудования

На АЗС реализована сложная схема подключения цистерн к заправочным колонкам, что обеспечивает гибкость в управлении топливными запасами. Каждая колонка может быть подключена к нескольким цистернам, а каждая цистерна может обслуживать несколько колонок. Это позволяет:

- 1) Распределять нагрузку между цистернами одного типа топлива
- 2) Обеспечивать бесперебойную работу при отключении одной из цистерн
- 3) Оптимизировать использование оборудования

1.3 Требования к функциональности

На основе анализа технического задания были выделены следующие функциональные требования к системе:

Таблица 1 – Функциональные требования к системе

Требование	Описание
Обслуживание клиентов	Продажа топлива с выбором колонки, типа топлива и количества литров
Управление цистернами	Контроль уровня топлива, автоматическое отключение при низком уровне, ручное управление
Пополнение запасов	Оформление поступления топлива от бензовоза
Статистика и отчеты	Учет продаж, доходов, количества обслуженных клиентов
История операций	Ведение журнала всех транзакций
Перекачка топлива	Перемещение топлива между цистернами одного типа
Аварийный режим	Блокировка всех операций при возникновении аварийной ситуации
Сохранение данных	Сохранение состояния системы между сеансами работы

2 Архитектура и реализация системы

2.1 Структура классов

Система построена на основе объектно-ориентированного подхода. Основные классы программы:

- 1) **Cistern (Цистерна)** - отвечает за хранение и управление запасами топлива
- 2) **FuelColumn (Заправочная колонка)** - управляет доступными типами топлива на колонке
- 3) **Transaction (Транзакция)** - представляет запись о совершенной операции
- 4) **Statistics (Статистика)** - накапливает и хранит данные о работе АЗС
- 5) **GasStation (АЗС)** - основной класс, объединяющий все компоненты системы

2.2 Ключевые алгоритмы

2.2.1 Алгоритм обслуживания клиента

- 1) Выбор заправочной колонки из доступных
- 2) Проверка доступных типов топлива на выбранной колонке
- 3) Выбор типа топлива и проверка доступности в подключенной цистерне
- 4) Ввод количества литров и проверка достаточности запаса
- 5) Расчет стоимости на основе установленных цен
- 6) Подтверждение операции и списание топлива
- 7) Фиксация транзакции и обновление статистики

2.2.2 Алгоритм управления цистернами

- 1) Регулярная проверка уровня топлива во всех цистернах
- 2) Автоматическое отключение цистерны при достижении минимального уровня
- 3) Предупреждение оператора о низком уровне топлива
- 4) Возможность ручного включения/отключения цистерн
- 5) Контроль при пополнении (недопущение переполнения)

2.3 Хранение данных

Для обеспечения сохранности данных между сеансами работы системы реализован механизм сериализации в формат JSON. Сохраняются следующие данные:

- 1) Состояние всех цистерн (текущий объем, состояние включения)
- 2) Статистика продаж и транзакций
- 3) Состояние аварийного режима

Данные сохраняются в файл `gas_station_data.json` при каждом изменении состояния системы.

3 Реализация основных функций

3.1 Модуль обслуживания клиентов

Реализация функции обслуживания клиента включает следующие этапы:

- 1) Проверка активности аварийного режима
- 2) Отображение списка доступных заправочных колонок
- 3) Получение выбора пользователя и проверка корректности
- 4) Определение доступных типов топлива на выбранной колонке
- 5) Проверка состояния подключенных цистерн
- 6) Ввод количества литров и проверка достаточности запаса
- 7) Расчет стоимости на основе установленных цен
- 8) Запрос подтверждения операции у пользователя
- 9) Списание топлива и обновление статистики
- 10) Фиксация транзакции в истории операций

3.2 Модуль управления цистернами

Класс `Cistern` реализует следующую логику управления:

- 1) Инициализация с параметрами: название, тип топлива, максимальный объем
- 2) Автоматическая установка начального уровня (50% от максимального)
- 3) Контроль минимального уровня (10% от максимального объема)
- 4) Проверка возможности отпуска топлива с учетом состояния и запасов
- 5) Автоматическое отключение при достижении минимального уровня
- 6) Контроль при пополнении (проверка на переполнение)
- 7) Сериализация и десериализация состояния для сохранения в файл

3.3 Модуль статистики и отчетности

Система ведет детальную статистику по всем аспектам работы АЗС:

- 1) Общий доход от продаж топлива
- 2) Количество обслуженных автомобилей
- 3) Детальная статистика по каждому типу топлива:

- Количество проданных литров
 - Полученный доход
 - Количество транзакций
- 4) История последних 100 операций с timestamp
- 5) Ограничение размера истории для предотвращения переполнения памяти

4 Тестирование и результаты

4.1 Методология тестирования

Для проверки работоспособности системы были проведены следующие виды тестирования:

- 1) **Функциональное тестирование** - проверка всех функций системы согласно требованиям
- 2) **Тестирование граничных условий** - работа с минимальными и максимальными значениями
- 3) **Тестирование обработки ошибок** - проверка реакции на некорректные данные
- 4) **Интеграционное тестирование** - проверка взаимодействия компонентов системы
- 5) **Тестирование восстановления данных** - проверка загрузки сохраненного состояния

4.2 Результаты тестирования

В результате тестирования подтверждена работоспособность всех основных функций системы:

Таблица 2 – Результаты тестирования системы

Функция	Описание теста	Результат
Обслуживание клиента	Продажа 20 литров АИ-95	Успешно
Проверка цистерн	Отображение состояния всех цистерн	Успешно
Пополнение топлива	Заправка цистерны на 5000 литров	Успешно
Статистика	Формирование отчета по продажам	Успешно
История операций	Просмотр последних транзакций	Успешно
Перекачка топлива	Перемещение 1000 литров между цистернами	Успешно
Аварийный режим	Активация и деактивация аварийного режима	Успешно
Сохранение данных	Сохранение и восстановление состояния	Успешно

4.3 Обработка исключительных ситуаций

Система корректно обрабатывает следующие исключительные ситуации:

- 1) Попытка продажи при отсутствии достаточного количества топлива
- 2) Попытка пополнения сверх максимальной емкости цистерны
- 3) Выбор несуществующей колонки или типа топлива
- 4) Ввод некорректных числовых значений
- 5) Отсутствие файла данных при запуске системы
- 6) Попытка перекачки топлива между цистернами разных типов
- 7) Активация аварийного режима при работающих операциях

Заключение

В ходе выполнения проекта была разработана полнофункциональная консольная система управления автозаправочной станцией, соответствующая всем требованиям технического задания. Система успешно решает следующие задачи:

- 1) Автоматизация процесса обслуживания клиентов с расчетом стоимости и учетом топлива
- 2) Эффективное управление запасами топлива в цистернах с автоматическим контролем уровня
- 3) Ведение полной статистики по всем аспектам работы АЗС
- 4) Обеспечение безопасности через аварийный режим и резервное копирование данных
- 5) Сохранение состояния системы между сеансами работы

Разработанная система демонстрирует следующие преимущества:

- 1) Удобный консольный интерфейс с пошаговым руководством пользователя
- 2) Надежное хранение данных в формате JSON
- 3) Модульная архитектура, позволяющая легко расширять функциональность
- 4) Полная обработка исключительных ситуаций
- 5) Соответствие реальным бизнес-процессам АЗС

Система может быть использована в учебных целях для демонстрации принципов объектно-ориентированного программирования, работы с файлами и построения консольных приложений. При необходимости система может быть расширена добавлением графического интерфейса, сетевых функций или интеграции с базами данных.

Перспективы развития системы включают:

- 1) Добавление модуля анализа эффективности работы АЗС
- 2) Реализация функций планирования поставок топлива
- 3) Интеграция с системами учета и отчетности
- 4) Разработка веб-интерфейса для удаленного управления
- 5) Добавление многопользовательского режима с разграничением прав доступа

Разработанная система является законченным программным продуктом, готовым к использованию в учебном процессе и в качестве основы для дальнейшего развития.

Список использованных источников

Исходный код программы

Основной модуль программы (`main.py`) содержит полную реализацию системы управления АЗС. Код структурирован и документирован в соответствии с требованиями к промышленному программированию.

Программа состоит из следующих основных компонентов:

- 1) Константы и настройки системы (типы топлива, цены, схема подключения)
- 2) Класс `Cistern` для управления цистернами
- 3) Класс `FuelColumn` для управления заправочными колонками
- 4) Класс `Transaction` для учета операций
- 5) Класс `Statistics` для ведения статистики
- 6) Класс `GasStation` как основной класс системы
- 7) Функции для работы с меню и пользовательским интерфейсом
- 8) Главная функция `main()` с основным циклом программы

Общий объем исходного кода составляет около 800 строк. Программа использует стандартные библиотеки Python: `json`, `datetime`, `os`, `typing`.

Примеры работы системы

Пример сеанса работы с системой управления АЗС включает следующие этапы:

- 1) Запуск программы и загрузка сохраненного состояния
- 2) Отображение главного меню с доступными функциями
- 3) Выбор функции обслуживания клиента
- 4) Пошаговое выполнение операции продажи топлива:
 - Выбор заправочной колонки (от 1 до 8)
 - Выбор типа топлива из доступных на выбранной колонке
 - Ввод количества литров
 - Подтверждение операции
 - Отображение результата и возврат в главное меню
- 5) Использование других функций системы:
 - Просмотр состояния цистерн
 - Оформление пополнения запасов
 - Просмотр статистики и истории операций
 - Управление цистернами (включение/отключение)
- 6) Завершение работы с сохранением состояния

Система обеспечивает интуитивно понятный интерфейс с подробными подсказками на каждом этапе работы. Все операции логируются и сохраняются для последующего анализа.