

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

ОТЧЕТ ПО КУРСОВОМУ ПРОЕКТУ

по дисциплине
«Программирование»

Студент

гр. БИН-25-3

Преподаватель

М.В. Кирийчук

М.В. Водяницкий

Техническое задание

0.1 *

Техническое задание - Система управления автозаправочной станцией

Вы работаете в российской компании "НефтеСофт" разрабатывающей программное обеспечение для автоматизации автозаправочных станций. Вам поручено создать консольный прототип системы управления АЗС, который используется операторами и техническим персоналом станции

Система должна учитывать реальные процессы работы заправки: продажи топлива, контроль запасов, обслуживание цистерн, аварийные ситуации и ведение статистики

0.2 *

1. Общая идея программы

Программа представляет собой консольную систему управления автозаправочной станцией, которая позволяет:

- обслуживать клиентов (касса)
- контролировать запасы топлива
- управлять цистернами и колонками
- оформлять пополнение топлива
- вести историю операций и статистику
- обрабатывать аварийные ситуации

Программа работает в виде меню с выбором действий и функционирует в непрерывном цикле до выхода пользователя

0.3 *

2. Топливо и цистерны

0.3.1 *

2.1 Типы топлива

На заправке используются следующие виды топлива:

- АИ-92
- АИ-95
- АИ-98
- ДТ (дизельное топливо)

0.3.2 *

2.2 Цистерны

- Для одного типа топлива может существовать несколько подземных цистерн
- Каждая цистерна имеет:
 - тип топлива
 - максимальный объем
 - текущий уровень топлива
 - состояние (включена / отключена)
 - минимальный допустимый уровень

Важно:

- Если уровень топлива в цистерне падает ниже минимального порога:
- цистерна автоматически отключается
- из нее нельзя отпускать топливо
- После пополнения:
 - цистерна не включается автоматически (после пополнения топливом)
 - включение производится вручную через меню

0.4 *

3. Колонки

0.4.1 *

3.1 Общая схема

- На заправке есть несколько колонок
- Каждая колонка:
 - поддерживает несколько типов топлива
 - каждый тип топлива (пистолет) подключен к конкретной цистерне
- Не все колонки подключены ко всем цистернам

0.4.2 *

3.2 Схема заправки

- Цистерны:
 - АИ-95 №1 → колонки 1 - 4
 - АИ-95 №2 → колонки 5 - 8
 - АИ-92 → колонки 1 - 6
 - АИ-98 → колонки 3 - 6
 - ДТ → колонки 3 - 8

- Всего колонок: 8
- Каждая колонка имеет 2 - 3 пистолета

0.5 *

4. Главное меню программы

В главном меню пользователь может выбрать одно из действий:

- 1) Обслужить клиента (касса)
- 2) Проверить состояние цистерн
- 3) Оформить пополнение топлива
- 4) Баланс и статистика
- 5) История операций
- 6) Перекачка топлива между цистернами
- 7) Включение / отключение цистерн
- 8) Состояние колонок
- 9) EMERGENCY - аварийная ситуация
- 0) Выход

0.6 *

5. Функциональные требования

0.6.1 *

5.1 Обслуживание клиента (касса)

Система должна:

- принять запрос:
- номер колонки
- тип топлива
- количество литров
- проверить:
- доступность колонки
- состояние цистерны
- достаточность топлива
- рассчитать стоимость
- запросить подтверждение оплаты
- при подтверждении:
- списать топливо

- зафиксировать транзакцию
- при нехватке топлива:
- сообщить об ошибке
- не допустить оплату

0.6.2 *

5.2 Проверка состояния цистерн

Вывод информации:

- тип топлива
- текущий объем
- максимальный объем
- состояние (включена / отключена)
- предупреждения о низком уровне

0.6.3 *

5.3 Пополнение топлива

При прибытии бензовоза:

- выбрать тип топлива
- выбрать конкретную цистерну
- указать количество литров
- проверить, не превышает ли объем максимум
- зафиксировать операцию в истории

0.6.4 *

5.4 Баланс и статистика

Система должна отображать:

- общий доход
- количество обслуженных автомобилей
- сколько раз покупали каждый тип топлива
- сколько литров каждого топлива продано
- доход по каждому виду топлива

0.6.5 *

5.5 История операций

Вывод последних операций:

- продажи топлива
- пополнение цистерн

- перекачка топлива
- отключения и включения цистерн
- аварийные события

0.6.6 *

5.6 Перекачка топлива

- Перекачка возможна только между цистернами одного типа топлива
- Указывается:
 - источник
 - приемник
 - объем
- Проверяется:
 - достаточность топлива
 - вместимость целевой цистерны

0.6.7 *

5.7 Управление цистернами

Пользователь может:

- вручную отключать цистерны
- вручную включать цистерны (если уровень топлива достаточен)

0.6.8 *

5.8 Состояние колонок

Выводится информация:

- номер колонки
- какие типы топлива доступны
- к каким цистернам они подключены
- какие пистолеты временно не работают (определяется по подключенным цистернам)

0.6.9 *

5.9 Аварийная ситуация (Emergency)

- При подтверждении аварии:
 - все цистерны блокируются
 - заправка прекращает работу
 - фиксируется аварийное событие
 - имитируется вызов аварийных служб

- Выход из аварийного режима возможен только вручную
- При выходе из аварийного режима цистерны автоматически НЕ разблокируются

0.7 *

6. Хранение данных

Все данные обязательно сохраняются в файлах (рекомендуется JSON):

- состояние цистерн
- схема колонок
- баланс и статистика
- история операций

Перезапуск программы не должен сбрасывать состояние заправки

0.8 *

7. Ограничения и требования

- Программа консольная
- Работа через меню и ввод команд
- Язык программирования не ограничен
- Код должен быть структурирован и читаем

1 Описание системы

Система управления автозаправочной станцией реализована на языке Python в объектно-ориентированном стиле. Основной класс `FuelStation` содержит всю логику работы АЗС.

1.1 Структура данных

- **Цистерны (cisterns)**: Словарь с данными о каждой цистерне (тип топлива, максимальный и текущий объём, состояние включения, минимальный уровень)
- **Колонки (pumps)**: Словарь с информацией о доступных видах топлива на каждой колонке
- **Баланс и статистика**: Учёт обслуженных автомобилей, проданного топлива и доходов по типам топлива
- **История операций**: Лог всех действий с временными метками
- **Аварийный режим**: Флаг для блокировки всех операций при аварии

1.2 Функциональные модули

1. Обслуживание клиентов (касса)
2. Мониторинг состояния цистерн
3. Пополнение запасов топлива
4. Просмотр баланса и статистики
5. Просмотр истории операций
6. Перекачка топлива между цистернами
7. Управление включением/отключением цистерн
8. Просмотр состояния колонок
9. Аварийный режим

2 Реализация

2.1 Архитектура системы

Класс `FuelStation` инкапсулирует все аспекты работы АЗС. Используются следующие основные методы:

- `__init__()` — инициализация начальных данных
- `initialize_data()` — настройка начального состояния цистерн и колонок
- `save_data()` / `load_data()` — сохранение и загрузка состояния в JSON
- `log_operation()` — запись действий в историю
- `check_low_levels()` — автоматическая проверка и отключение цистерн с низким уровнем

2.2 Меню и интерфейс

Система предоставляет текстовое меню с 9 пунктами управления. При запуске автоматически проверяются цистерны с низким уровнем топлива и выводятся соответствующие предупреждения.

2.3 Безопасность и контроль

- Автоматическое отключение цистерн при достижении минимального уровня
- Проверка достаточности топлива при обслуживании клиентов
- Валидация вводимых данных
- Блокировка операций в аварийном режиме
- Подтверждение критических операций

3 Примеры работы

3.1 Обслуживание клиента

Пример работы системы при обслуживании клиента:

--- Обслуживание клиента ---

Доступные колонки:

- 1) Колонка 1
- 2) Колонка 2
- 3) Колонка 3
- 4) Колонка 4

Выберите колонку: 3

Колонка 3

Доступные виды топлива:

- 1) AI-95 (цистерна AI-95_1)
- 2) AI-92 (цистерна AI-92_1)
- 3) AI-98 (цистерна AI-98_1) - НЕДОСТУПНО
- 4) DT (цистерна DT_1)

Выберите тип топлива: 1

Введите количество литров: 50

Стоимость:

50 л × 51.20 ₺ = 2560.00 ₺

Подтвердить оплату? (y/n): y

Операция выполнена успешно.

Спасибо за покупку!

3.2 Проверка состояния цистерн

Пример вывода информации о состоянии цистерн:

--- Состояние цистерн ---

Доступные цистерны:

AI-92_1		12400	/	20000 л		ВКЛ
AI-95_1		9800	/	20000 л		ВКЛ
AI-95_2		1200	/	20000 л		ВЫКЛ (ниже порога)
AI-98_1		10000	/	15000 л		ВЫКЛ

DT_1 | 15600 / 25000 л | ВКЛ

3.3 Аварийный режим

Пример работы аварийного режима:

--- EMERGENCY - аварийная ситуация ---

ВНИМАНИЕ! Все цистерны будут заблокированы!

Подтвердить аварию? (y/n): y

Аварийный режим активирован. Заправка остановлена.

Выйти из аварийного режима? (y/n): y

Аварийный режим завершен.

4 Исходный код

В этом разделе представлены ключевые фрагменты исходного кода системы.

4.1 Класс FuelStation и инициализация

```

1 import json
2 import os
3 from datetime import datetime
4
5 class FuelStation:
6     def __init__(self):
7         self.cisterns = {}
8         self.fuel_prices = {
9             'AI-92': 47.50,
10            'AI-95': 51.20,
11            'AI-98': 58.30,
12            'DT': 56.00
13        }
14        self.pumps = {}
15        self.balance = 0.0
16        self.stats = {
17            'cars_served': 0,
18            'fuel_sold': {'AI-92': 0, 'AI-95': 0, 'AI-98': 0, 'DT': 0},
19            'income_by_fuel': {'AI-92': 0, 'AI-95': 0, 'AI-98': 0, 'DT': 0}
20        }
21        self.history = []
22        self.emergency_mode = False
23
24        # Инициализация начальных данных
25        self.initialize_data()
26
27    def initialize_data(self):
28        # Инициализация цистерн
29        self.cisterns = {
30            'AI-92_1': {
31                'type': 'AI-92',
32                'max_volume': 20000,
33                'current_volume': 12400,
34                'enabled': True,
35                'min_level': 1000
36            },
37            'AI-95_1': {
38                'type': 'AI-95',
39                'max_volume': 20000,
40                'current_volume': 9800,
41                'enabled': True,
42                'min_level': 1000
43            },
44            'AI-95_2': {
45                'type': 'AI-95',
46                'max_volume': 20000,
47                'current_volume': 1200,
48                'enabled': False,
49                'min_level': 1000
50            },
51            'AI-98_1': {
52                'type': 'AI-98',
53                'max_volume': 15000,
54                'current_volume': 10000,
55                'enabled': False,
56                'min_level': 1000
57            },
58            'DT_1': {
59                'type': 'DT',
60                'max_volume': 25000,

```

Рисунок 1 – Листинг класса FuelStation

4.2 Метод обслуживания клиента

```

1  Инициализация начальных данных Инициализация цистерн Инициализация колонок Автоматически отключаем если уровень ниже порога
2  def display_menu(self):
3      print("=" * 40)
4      print("\A3C СеверНефть<<>>")
5      print("Система управления заправочной станцией")
6      print("=" * 40)
7
8      low_level_cisterns = self.check_low_levels()
9      if low_level_cisterns:
10         print("\ВНИМАНИЕ!")
11         print("Обнаружены отключённые цистерны:")
12         for cistern_id in low_level_cisterns:
13             cistern = self.cisterns[cistern_id]
14             print(f" - {cistern_id} низкий( уровень топлива)")
15
16         print("-" * 40)
17         print("Выберите действие:")
18         print("1) Обслужить клиента касса()")
19         print("2) Проверить состояние цистерн")
20         print("3) Оформить пополнение топлива")
21         print("4) Баланс и статистика")
22         print("5) История операций")
23         print("6) Перекачка топлива между цистернами")
24         print("7) Включение / отключение цистерн")
25         print("8) Состояние колонок")
26         print("9) EMERGENCY - аварийная ситуация")
27         print("0) Выход")
28
29     def serve_customer(self):
30         print("--- Обслуживание клиента ---")
31
32         if self.emergency_mode:
33             print("Ошибка: Станция находится в аварийном режиме!")
34             input("Нажмите Enter для возврата в меню...")
35             return
36
37         print("Доступные колонки:")
38         for pump_num in sorted(self.pumps.keys()):
39             available_fuels = []
40             for fuel_type, cistern_id in self.pumps[pump_num]['fuels'].items():
41                 if self.cisterns[cistern_id]['enabled']:
42                     available_fuels.append(fuel_type)
43
44             if available_fuels:
45                 print(f"{pump_num}) Колонка {pump_num}")
46
47         try:
48             pump_choice = int(input("Выберите колонку: "))
49             if pump_choice not in self.pumps:
50                 print("Неверный номер колонки!")
51                 input("Нажмите Enter для возврата в меню...")
52                 return
53
54             pump = self.pumps[pump_choice]
55             available_fuels = {}
56             fuel_options = []
57
58             print(f"Колонка {pump_choice}")
59             print("Доступные виды топлива:")
60
61             option_num = 1
62             for fuel_type, cistern_id in pump['fuels'].items():
63                 cistern = self.cisterns[cistern_id]
64                 if cistern['enabled']:
65                     available_fuels[option_num] = (fuel_type, cistern_id)
66                     print(f"{option_num}) {fuel_type} цистерна( {cistern_id})")
67                     option_num += 1
68                 else:
69                     print(f"{option_num}) {fuel_type} цистерна( {cistern_id}) - НЕДОСТУПНО")
70                     option_num += 1
71
72             fuel_choice = int(input("Выберите тип топлива: "))

```

Рисунок 2 – Листинг метода serve_customer