



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение*

*Высшего образования*

«МИРЭА - Российский технологический университет»

РТУ МИРЭА

---

Институт информационных технологий (ИИТ)

Кафедра Математического обеспечения и стандартизации

информационных технологий (МОСИТ)

Техническое задание программного продукта

Выполнили студенты: Соловьева М.А

Шпагина Ю.Н

Карасев Е.В

Группа:

ИКБО-43-23

Москва 2025

# СОДЕРЖАНИЕ

1 Введение .....	3
2 Основания для разработки.....	3
3 Назначение разработки .....	4
4 Требования к программе .....	4
4.1 Функциональные требования .....	4
4.2 Требования к надёжности .....	4
4.3 Требования эксплуатации .....	6
4.4 Требования к совместимости.....	7
5 Требования к интерфейсу .....	7
6 Критерии приёмки .....	8
7 Требования к документации .....	8
8 Порядок контроля и приёмки .....	9
9 Этапы и сроки разработки .....	10

## **1 Введение**

Программный продукт «MiniShop» представляет собой консольное приложение для оформления простых заказов.

Назначение — обучение тестированию методом «черного ящика»: проверка поведения системы на основе входов/выходов.

## **2 Основания для разработки**

Разработка выполняется в рамках учебной **\*\*Практической работы №1\*\*** по тестированию ПО методом «черного ящика».

Нормативные ссылки: ГОСТ 19.201-78, ГОСТ 34.602-2020, ISO/IEC/IEEE 29148:2018.

### **3 Назначение разработки**

Система позволяет создать заказ из нескольких позиций (название, цена, количество, промокод) и получить итог к оплате, сохранив сведения в CSV.

Учебная цель — предоставить тестовый стенд с намеренно посеянными дефектами.

### **4 Требования к программе**

#### **4.1 Функциональные требования**

- Ввод позиций заказа (название, цена, количество, промокод).
- Расчет итога с учетом НДС и скидочного кода.
- Просмотр текущего «чека» и суммы.
- Сохранение заказа в `orders.csv`.
- Консольное меню управления.

#### **4.2 Требования к надёжности**

- Приложение должно сохранять заказ на диск без зависимостей от СУБД.
- При ошибочном вводе должен выводиться текст ошибки и запрашиваться повторный ввод (фактическое поведение может отличаться — объект тестирования).

### **4.3 Требования эксплуатации**

- ОС: Windows/Linux/macOS; интерпретатор Python 3.8+.
- Консоль/терминал, раскладка RU/EN.
- Локаль — по умолчанию (возможны особенности с запятой/точкой)

для бизнеса операции (оформление заказа, оплата) должны сохранять работоспособность при: задержке сети до 300 мс, потере пакетов до 2%.

#### **4.4 Требования к совместимости**

- Не требует внешних библиотек.
- Файл `orders.csv` совместим с табличными редакторами (Excel/LibreOffice).

#### **5 Требования к интерфейсу**

- Простой текстовый интерфейс меню (1–4).
- Все суммы отображаются в рублях с указанием валюты.

## **6 Критерии приёмки**

- Не менее 95% тест-кейсов должны завершаться ожидаемым результатом для исправленной версии приложения.
- Сохранение должно формировать корректные строки, пригодные для импорта в табличные редакторы.
- Итоговая сумма должна корректно рассчитываться для валидных входов согласно разделу 4.1 (см. формулы в TestPlan).

## **7 Требования к документации**

- Руководство пользователя (User\_Manual.md)
- Тест-план и набор тест-кейсов (TestPlan.md)
- Отчет о найденных дефектах другой команды (будет добавлен после обмена)
- Итоговый отчет по работе (Report\_Practice\_1.md)

## **8 Порядок контроля и приёмки**

- Функциональное тестирование методом «черного ящика» (эквивалентные классы, граничные значения, негативные сценарии).
- Проверка корректности документации (ревью, чек-листы).
- Приемочные испытания: запуск, ввод типовых сценариев, сверка результатов с ожидаемыми значениями.

## **9 Этапы и сроки разработки**

- Инициация/планирование — 0.5 дн.

- Реализация прототипа — 1 дн.
- Подготовка документации — 1 дн.
- Внутреннее тестирование — 0.5 дн.
- Передача на внешнее тестирование — 0.5 дн.