



МИНОБРНАУКИ РОССИИ

*Федеральное государственное бюджетное образовательное учреждение  
высшего образования*

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

---

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных  
технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1**

**по дисциплине**

**«Тестирование и верификация программного обеспечения»**

**Тема:** Тестирование программного продукта методом “черного ящика”

Выполнили студенты группы ИКБО-66-23

Команда BratLand  
Лобачев Е.К.  
Елисеев И.А.  
Мельник К.Н.  
Силютин Н.С.

Практическая работа выполнена

«\_\_» \_\_\_\_\_ 202\_\_ г.

(подпись студента)

«Зачтено»

«\_\_» \_\_\_\_\_ 202\_\_ г.

(подпись руководителя)

## 1. Введение

Настоящее техническое задание определяет цели, требования и порядок разработки консольного приложения «Конвертер валют» (далее — Программа). Программа реализуется на языке Kotlin и предназначена для выполнения операций по переводу денежных сумм между различными валютами.

Приложение будет использоваться в учебных целях для демонстрации принципов разработки и последующего тестирования программного обеспечения методом «чёрного ящика».

---

## 2. Основания для разработки

Разработка проводится в рамках практической работы по дисциплине «Тестирование и верификация программного обеспечения».

Основания для создания продукта:

- необходимость предоставления исполняемого приложения для тестирования другой командой;
  - закрепление навыков составления технической документации в соответствии с ГОСТ;
  - отработка методологии «чёрного ящика».
- 

## 3. Назначение разработки

Программа предназначена для пользователей, которым требуется быстрый инструмент для перевода денежных сумм между несколькими валютами (рубль, доллар США, евро, юань).

Основные задачи разработки:

- реализовать простой и удобный интерфейс;
  - предоставить возможность конвертации валют в обоих направлениях;
  - отработать процесс тестирования ПП с внесёнными ошибками.
- 

## 4. Требования к программе

### 4.1 Функциональные требования

Программа должна обеспечивать:

1. Ввод исходной суммы в выбранной валюте.
2. Выбор валюты для перевода.
3. Отображение результата пересчёта в целевой валюте.
4. Возможность повторного ввода без перезапуска программы.
5. Поддержку не менее четырёх валют: RUB, USD, EUR, CNY.

#### 4.2 Требования к надёжности

- При некорректном вводе программа должна выводить сообщение об ошибке без аварийного завершения.
- Результаты вычислений должны быть воспроизводимыми при одинаковом вводе.

#### 4.3 Требования к условиям эксплуатации

- Операционная система: Windows, Linux, macOS.
- Необходим установленный Kotlin (версии 1.8 и выше) или JVM (версии 11 и выше).

#### 4.4 Требования к совместимости

Программа должна запускаться в консоли и не требовать сторонних библиотек.

#### 4.5 Требования к интерфейсу

- Текстовое меню выбора исходной и целевой валюты.
- Поле для ввода суммы.
- Отображение результата в формате:  
100 RUB = 1.05 USD

---

### 5. Критерии приёмки

Программа считается принятой, если:

1. Реализованы все функции из раздела 4.1.
  2. Программа корректно выполняет пересчёт валют и выдаёт результат без критических ошибок.
  3. Интерфейс соответствует заявленным требованиям.
  4. Программа устойчива к некорректному вводу данных.
-

## 6. Порядок контроля и приёмки

Контроль качества осуществляется методом функционального тестирования («чёрного ящика»).

- **Этап 1:** предоставление исполняемого файла и инструкции по запуску.
- **Этап 2:** проведение тестирования: ввод различных сумм, проверка корректности пересчёта, реакция на ошибки ввода.
- **Этап 3:** фиксация найденных дефектов в отчёте.

---

## 7. Этапы и сроки разработки

№	Этап разработки	Срок выполнения	Примечание
1	Проектирование структуры ПП	1 день	Определение функционала
2	Реализация базовой логики	2 дня	Ввод/вывод, пересчёт валют
3	Добавление пользовательского меню	1 день	Навигация и интерфейс
4	Тестирование и отладка	1 день	Проверка корректности работы
5	Внесение преднамеренных ошибок	1 день	Для целей тестирования
6	Подготовка документации	1 день	Инструкция и описание
	<b>Итого:</b>	<b>7 дней</b>	—

## Документация на программный продукт «Конвертер валют»

### 1. Инструкция по запуску

1. Установите Kotlin версии 1.8+ или Java (JVM 11+).
2. Скачайте исходный файл CurrencyConverter.kt.
3. Скомпилируйте программу:
4. `kotlinc CurrencyConverter.kt -include-runtime -d CurrencyConverter.jar`
5. Запустите программу:

6. java -jar CurrencyConverter.jar

---

## 2. Инструкция по использованию

После запуска программы в консоли отобразится меню:

1. Введите сумму, которую хотите конвертировать.
  2. Выберите исходную валюту (RUB, USD, EUR, CNY).
  3. Выберите целевую валюту.
  4. Программа выведет результат конвертации в формате:
  5. 100 RUB = 1.05 USD
  6. Для повторного ввода просто введите новые данные.
- 

## Список преднамеренно внесённых ошибок в ПО

В целях выполнения практической работы в программу были внесены следующие ошибки:

### 1. Ошибка округления

Результат конвертации всегда округляется до **целого числа**, что приводит к потере точности.

**Как обнаружить:** конвертировать 1 USD в RUB и обратно. Результат не совпадёт.

---

### 2. Ошибка обработки ввода

Если пользователь вводит не число (например, «abc»), программа завершается с ошибкой вместо отображения сообщения.

**Как обнаружить:** ввести «abc» вместо суммы.

---

### 3. Неверный курс валюты

Курс EUR → RUB указан неправильно (например, 1 EUR = 50 RUB вместо ~100).

**Как обнаружить:** сравнить результаты пересчёта с реальными курсами.

---

### 4. Ошибка направления пересчёта

Конвертация RUB → USD работает, а обратное преобразование (USD → RUB) выдаёт неверный результат.  
**Как обнаружить:** перевести 100 RUB в USD, затем обратно. Итоговая сумма ≠ 100 RUB.

---

## 5. Отсутствие сохранения выбора

Программа каждый раз сбрасывает валюты по умолчанию (RUB → USD), даже если пользователь выбрал другие.  
**Как обнаружить:** выполнить несколько переводов подряд.

---

## 6. Ошибка интерфейса

Результат выводится с **избыточным количеством знаков** после запятой (например, 1.0000000001 USD).  
**Как обнаружить:** перевести дробную сумму (например, 123.45 RUB в USD).

---

## 7. Ошибка завершения работы

В программе отсутствует команда выхода. Для остановки требуется **принудительное завершение** (Ctrl+C).  
**Как обнаружить:** попробовать завершить работу через меню.