



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)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1**  
по дисциплине «Тестирование и верификация программного обеспечения»

**Тема:** Тестирование программного продукта методом «черного ящика»

**Название команды:** ИКБО-66-23-ForestBeavers

**Состав команды:** Маркелов М.Д.  
Волков А.В.  
Ян Х.  
Цибизов М.А.

**Дата выполнения:** 9 сентября 2025 г.

Москва 2025

## ***Часть 1. Разработка технического задания и программного продукта***

### **1. Техническое задание (ТЗ) собственного ПП**

#### **1.1 Введение**

Настоящее техническое задание определяет цели, требования и этапы разработки десктопного приложения с графическим интерфейсом "Calculator" (далее — "Программа"). Проект реализуется на языке Golang с использованием библиотек bufio, fmt, os, strconv, strings и предназначен для удобного управления списком персональных задач.

#### **1.2 Основания для разработки**

Разработка проводится в рамках самостоятельного обучения разработчика. Основанием является личная инициатива с целью:

- Практического применения знаний Golang.
- Создания законченного рабочего проекта для портфолио.
- Изучения принципов структурирования кода и разработки приложений.

#### **1.3 Назначения работника**

Программа предназначена для пользователей, которым требуется простой и быстрый инструмент для математических вычислений.

#### **1.4 Требование к программе**

- Операция сложения.
- Операция вычитания.
- Операция деления.
- Операция умножения.

##### **1.4.2 Требования к надежности**

- Программа не должна завершаться с ошибкой при вводе некорректных данных.

### 1.4.3 Требования к эксплуатации

- ОС: Windows, Linux, macOS.
- Необходим интерпретатор Golang версии 1.19.0 или выше.
- Необходимы библиотеки: bufio, fmt, os, strconv, string.

### 1.4.4 Требования к совместимости

Приложение должно быть кроссплатформенным.

Совместимость с операционными системами: • Приложение должно работать под управлением любых операционных систем, на которых могут быть запущены современные веб-браузеры (включая Windows, macOS, Linux). Это обеспечивается тем, что ПО заранее собрано под все ОС.

### 1.5 Требования к интерфейсу

- Графический интерфейс (CLI), реализованный на Golang.
- Поле ввода для новой задачи.
- Поле вывода результата.
- Инструкция пользования приложения
- Строка статуса с статистикой.

## 1.6 Требования приемки

Проект считается завершенным и успешным, если:

1. Все функции, перечисленные в п. 1.4, реализованы и работают корректно.
2. Программа работает стабильно, без критических ошибок.
3. Интерфейс интуитивно понятен и соответствует требованиям.
4. Код читаем, хорошо структурирован и содержит комментарии.

## 1.7 Порядок контроля приемки

Процедура контроля и приемки выполняется на основании формализованного процесса верификации, описанного в данном ТЗ.

1. **Метод тестирования:** Тестирование калькулятора проводится **методом черного ящика**. Фокус направлен на проверку корректности вычислений и обработки ошибок через пользовательский интерфейс, без доступа к исходному коду.
2. **Цель тестирования:** Проверить соответствие готового продукта всем функциональным и нефункциональным требованиям, изложенным в разделах 1.4 и 1.5 настоящего ТЗ.
3. **Процедура приемки:**
  - **Шаг 1:** Предоставление готового программного продукта и инструкции по запуску.
  - **Шаг 2:** Проведение приемочного тестирования по следующим критериям:
    - Соответствие функциональным требованиям (п. 1.4).
    - Соответствие требованиям к интерфейсу (п. 1.4).
    - Отсутствие критических ошибок (завершение работы, потеря данных).
  - **Шаг 3:** Фиксация результатов тестирования в виде отчета о тестировании.

4. **Критерий успешного прохождения приемки:** Успешное выполнение всех тестовых сценариев, составленных на основе требований данного ТЗ, и отсутствие критических дефектов.

функциональным и нефункциональным требованиям, изложенным в разделах 1.4 и 1.5 настоящего ТЗ.

Процедура приемки:

1. Предоставление ПП и инструкции по запуску
2. Проведение приемочного тестирования
3. Фиксация результатов.

## 1.8 Этапы и сроки разработки

№	Этап разработки	Срок исполнения	Примечание
1	Проектирование интерфейса	1 день	
2	Реализация базовой логики	1 день	
3	Реализация работы с файлами	2 дня	
4	Тестирование и отладка	1 день	
5	Тестирование и отладка	1 день	
6	Рефакторинг кода, комментирование	1 день	
7	Написание документации	1 день	
	Итого:	7 дней	<i>Сроки ориентировочные</i>

## 2. Дополнительная документация на собственный ПП

### 2.1 Инструкция по запуску

1. Убедитесь что запускаемый файл с уточнением системы запускается на соответствующей системе.

## 2.2 Инструкции по управлению

- **Добавление задачи:** Введите текст в верхнее поле ввода и нажмите Enter.
- **Выход из программы:** Напишите в командную строку слово "exit".

## 3. Описание ошибок внесенных в собственное ПО

В целях выполнения практической работы №1 в ПО были внесены следующие ошибки:

1. Ошибки математических операций: перепутаны мат. операции

*Листинг 1. Описание добавленной ошибки*

```
case "+":
    result = num1 - num2 // Неправильная
    математическая операция
case "-":
    result = num1 + num2 // Неправильная
    математическая
case "*":
    result = num1 / num2 // Неправильная
    математическая операция
case "/":
    result = num1 * num2 // Неправильная
    математическая операция
```

2. Ошибка математической операции вычитания: При операции вычитания выполняется операция сложения и добавляется число 2.

*Листинг 2. Описание добавленной ошибки*

```
case "-":
    result = num1 + num2 + 2 // Неправильная
```

3. Ошибка математической операции возведение в степень: При операции возведения в степень число возводится в  $n+1$ .

*Листинг 3. Описание добавленной ошибки*

```
case "^":  
    result = math.Pow(num1, (num2 + 1)) //  
    Неправильная вычисление квадрата числа
```

4. Ошибка математической операции: Отсутствие проверки деления на 0.

*Листинг 4. Описание добавленной ошибки*

```
case "/": // Отсутствие проверки при делении на 0
```

## ***Часть II. Тестирование ТЗ***

1. В пункте 1.4.3 предоставленного ТЗ неправильно указана рабочая версия Python. ПП не запускается на Python 3.9.6, что не соответствует ТЗ другой команды. По ТЗ программа должна запускаться на Python с версии 3.6.
2. В пункте 1.4.3 не указана нужная версия библиотеки Tkinter.
3. В пункте 1.4.4 указано, что исполняемой средой является браузер, а не ОС, что не является действительностью
4. В пункте 2.2 ТЗ соответствует всем требованиям.

## ***Часть III. Тестирование ПП***

### **1. Ошибки в продукте другой команды.**

**Описание бага №1:** При повреждение или неверном формате JSON файла, задачи не добавляются в программу.

```
todomaster > {} tasks.json > ...
1  [
2
3      "id": 0,
4      "text": "фываоыфжва",
5      "completed": true,
6      "due_date": null
7  ],
8  {
9      "id": 1,
10     "text": "lk;jsd;lkfs",
11     "completed": false,
12     "due_date": null
13 },
14 {
15     "id": 2,
16     "text": "s;dlkfjs;d",
17     "completed": false,
18     "due_date": null
19 },
```

Рисунок 1 — Демонстрация бага №1.

**Описание бага №2:** Неинклюзивный интерфейс — отсутствие кнопки «Добавить» для новой задачи, привязанно к «Enter».



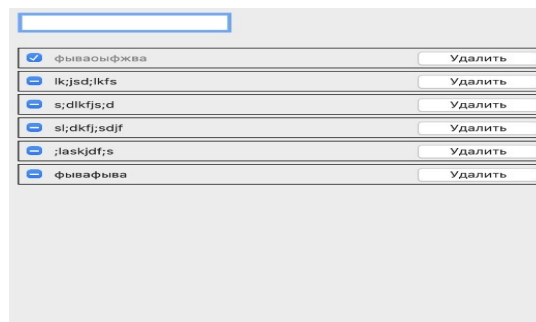


Рисунок 2 – Демонстрация бага №2.

**Описание бага №3:** Статус бар (отображение статистики) не обновляется при добавлении, удалении или изменении статуса задачи.

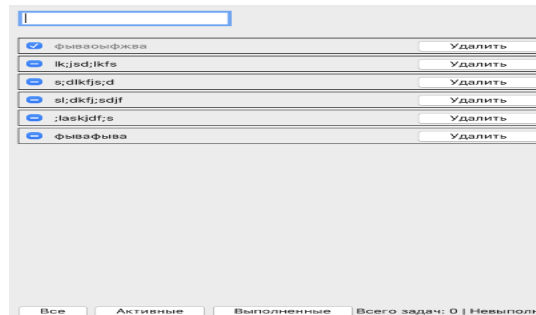


Рисунок 2 – Демонстрация бага №2

**Описание бага №4:** Поле `due_date` (срок выполнения) объявлено в структуре задачи, но полностью отсутствует функционал для его установки, редактирования и проверки.

```
{  
  "id": 3,  
  "text": "sl;dkfj;sdf",  
  "completed": false,  
  "due_date": null  
},
```

Рисунок 4 – Демонстрация бага 4.

**Описание бага №5:** Удаление задачи происходит по `id`, но `id` назначаются по порядковому номеру (индексу) в списке. После удаления одной

задачи `id` остальных задач перестают соответствовать их позиции в списке, что может привести к ошибкам.

```
[  
  {  
    "id": 1,  
    "text": "123",  
    "completed": false,  
    "due_date": null  
  },  
  {  
    "id": 2,  
    "text": "333",  
    "completed": false,  
    "due_date": null  
  },  
  {  
    "id": 2,  
    "text": "123",  
    "completed": false,  
    "due_date": null  
  }  
]
```

Рисунок 5 – Демонстрация бага 5.

## **Заключение**

Изучение технического задания и сопроводительной документации позволило выявить ряд замечаний и потенциальных проблем, связанных с неполным описанием функционала, недостаточной проработкой механизмов защиты данных и возможными сценариями возникновения ошибок. Для минимизации рисков на этапе разработки и эксплуатации необходимо доработать ТЗ, детализировав функциональные требования, тестовые сценарии и меры по обеспечению безопасности приложения.