|  |
| --- |
| 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**](https://online-edu.mirea.ru/course/index.php?categoryid=54) **(МОСИТ)**

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ**

по дисциплине «Тестирование и верификация программного обеспечения»

Команда «Три кота»

Состав: Ерхова В. А, Павлов Н. С., Хохряков А. Ю.

**Практическое занятие № 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Студент группы ИНБО-01-17 | *ИКБО-50-23, Павлов Н.С.* | (подпись) | |
| Преподаватель | *Ильичев Г.П* | (подпись) | |
| Отчет представлен | «12» сентября 2025 г. | |  | |

Москва 2025 г.

**Введение**

Программа "Конвертер величин" представляет собой desktop-приложение для быстрого и удобного преобразования различных физических величин между единицами измерения. Приложение предназначено для использования в образовательных целях, в быту, а также в профессиональной деятельности, где требуется оперативный пересчет значений между различными системами измерений.

Области применения:

* Учебный процесс в школах и вузах
* Инженерные расчеты
* Научно-исследовательская деятельность
* Повседневное использование для бытовых расчетов

**Основания для разработки**

Разработка программы осуществляется в рамках учебного проекта для демонстрации практических навыков программирования на Python с использованием библиотеки tkinter для создания графического интерфейса.

Исходные документы:

* Задание на практическую работу по программированию
* Рекомендации по разработке пользовательских интерфейсов

**Назначение разработки**

Программа предназначена для автоматизации процесса конвертации физических величин, что позволит:

* Сократить время расчета по сравнению с ручными вычислениями
* Исключить ошибки при пересчете значений между системами измерений
* Обеспечить наглядное представление результатов преобразования
* Предоставить удобный интерфейс для работы с различными категориями величин

**Требования к программе**

**Функциональные требования**

* Конвертация величин в категориях: длина, масса, температура, время, объем, площадь
* Возможность выбора конкретных единиц измерения внутри каждой категории
* Функция обмена выбранных единиц измерения
* Отображение результатов преобразования с точностью до 8 знаков после запятой
* Обработка ошибок ввода и некорректных данных
* Валидация вводимых значений (проверка на отрицательные значения для определенных категорий)
* Очистка всех полей ввода по запросу пользователя

**Требования к надежности**

* Обработка исключительных ситуаций без аварийного завершения программы
* Корректное восстановление после ошибок ввода
* Защита от некорректных операций (деление на ноль, нечисловые значения)
* Стабильная работа при непрерывном использовании

**Условия эксплуатации**

* Операционная система: Windows 7/8/10/11, Linux, MacOS
* Минимальные требования: 512 МБ ОЗУ, 50 МБ свободного места на диске
* Обязательное наличие интерпретатора Python версии 3.6 и выше
* Наличие библиотеки tkinter в составе Python

**Требования к совместимости**

* Совместимость с операционными системами семейства Windows, Linux, MacOS
* Поддержка кодировки UTF-8 для корректного отображения символов
* Независимость от дополнительного программного обеспечения (кроме Python)

**Требования к интерфейсу**

**Основные элементы интерфейса**

* Выпадающий список для выбора категории величин
* Поле ввода исходного значения с возможностью использования точки или запятой как разделителя
* Два выпадающих списка для выбора единиц измерения (исходная и целевая)
* Кнопка обмена единицами измерения (↔)
* Кнопка "Конвертировать" зеленого цвета с белым текстом
* Поле вывода результата с зеленым текстом на светлом фоне
* Область дополнительной информации о результате преобразования
* Кнопка "Очистить все" красного цвета с белым текстом

**Особенности интерфейса**

* Все элементы управления должны быть подписаны на русском языке
* Размер окна фиксированный (500x650 пикселей)
* Шрифты: Arial различных размеров (9-16) и начертаний
* Цветовая схема: сочетание зеленых, серых и белых тонов
* Кнопка конвертации должна быть заметным акцентным элементом
* Поле результата должно визуально отличаться от поля ввода

**Критерии приемки**

* Успешное выполнение 100% тестов
* Отсутствие аварийных завершений программы при любых сценариях использования
* Соответствие времени отклика интерфейса требованиям (не более 0.5 сек)
* Полное соответствие функциональности заявленным требованиям

**Требования к документации**

**Обязательная документация**

* Запрос пользователя с описанием требуемой системы
* Техническое описание программы
* Техническое задание
* Исходный код с соответствующими комментариями

**Порядок контроля и приемки**

**Методы тестирования**

* Функциональное тестирование всех элементов интерфейса
* Тестирование корректности преобразования величин
* Тестирование обработки ошибок и исключительных ситуаций
* Тестирование пользовательского интерфейса на удобство использования

**Приемочные испытания**

* Проверка на соответствие техническому заданию
* Тестирование на различных операционных системах
* Проверка работы при различных разрешениях экрана
* Оценка удобства интерфейса целевой аудиторией

**Этапы и сроки разработки**

1. Проектирование архитектуры приложения – 1 день
2. Разработка графического интерфейса – 2 дня
3. Реализация логики преобразования величин – 3 дня
4. Реализация обработки ошибок и исключений – 1 день
5. Тестирование и отладка программы – 2 дня
6. Написание документации – 1 день

Общий срок разработки: 10 рабочих дней

**Дополнительная документация на программный продукт "Конвертер величин"**

**1. Руководство пользователя**

**Обзор приложения**

"Конвертер величин" — это графическое приложение для конвертации различных физических единиц измерения. Оно поддерживает категории: Длина, Масса, Температура, Время, Объем и Площадь. Приложение позволяет вводить значение, выбирать исходную и целевую единицы, выполнять конвертацию и отображать результат.

**Установка и запуск**

* **Требования:** Python 3.x с установленной библиотекой Tkinter (обычно входит в стандартную поставку Python).
* **Запуск:** Сохраните код в файл main.py и выполните команду python main.py в терминале. Приложение откроется в отдельном окне.

**Интерфейс пользователя**

* **Категория:** Выберите категорию из выпадающего списка (например, "Длина").
* **Исходное значение:** Введите числовое значение в поле ввода. Поддерживаются целые и дробные числа (используйте точку или запятую как разделитель).
* **Из:** Выберите исходную единицу измерения.
* **В:** Выберите целевую единицу измерения.
* **Кнопка "↔":** Обменяет исходную и целевую единицы.
* **Кнопка "Конвертировать":** Выполнит расчет и отобразит результат.
* **Раздел результата:** Показывает конвертированное значение и дополнительную информацию (например, "10 метры = 1000 сантиметры").
* **Кнопка "Очистить все":** Сбросит все поля.

**Примеры использования**

1. **Конвертация длины:** Выберите "Длина", введите "1" в поле значения, "километры" в "Из", "метры" в "В". Нажмите "Конвертировать". Результат: "1000".
2. **Конвертация температуры:** Выберите "Температура", введите "0", "°C" в "Из", "°F" в "В". Результат: "32".
3. **Обмен единиц:** Нажмите "↔" для быстрого переключения между "Из" и "В".
4. **Обработка ошибок:** Если значение не число, приложение покажет сообщение "Введите корректное число". Для отрицательных значений в "Масса" или "Длина" — "Значение не может быть отрицательным".

**Советы**

* Используйте клавишу Enter для конвертации после ввода значения.
* Результат форматируется для удобства: с разным количеством знаков после запятой в зависимости от величины.
* Для температуры учитывается абсолютный ноль (в Кельвинах результат не может быть отрицательным).

**2. Описание архитектуры системы**

**Общая структура**

Приложение построено на основе объектно-ориентированного подхода. Основной класс UnitConverter инкапсулирует всю логику: инициализацию данных, настройку UI и обработку конвертации. Главная функция main() создает корневое окно Tkinter и экземпляр класса.

* **Модули и библиотеки:**
  + tkinter и ttk для графического интерфейса.
  + messagebox для вывода ошибок.
* **Данные:**
  + Словарь units хранит списки единиц по категориям.
  + Словарь conversion\_factors содержит коэффициенты конвертации (относительно базовой единицы, первой в списке).
* **Методы:**
  + \_\_init\_\_: Инициализирует окно, данные и UI.
  + setup\_ui: Настраивает элементы интерфейса (фреймы, метки, комбобоксы, кнопки).
  + update\_units: Обновляет списки единиц при смене категории.
  + swap\_units: Обменивает единицы "Из" и "В".
  + clear\_fields: Очищает поля ввода и результата.
  + convert\_temperature: Специальная функция для температуры (конвертация через Цельсий).
  + convert: Основная логика конвертации, включая проверки и форматирование.

**Архитектурные принципы**

* **Модульность:** UI отделен от логики конвертации.
* **Событийно-ориентированный подход:** Используются привязки событий (e.g., <<ComboboxSelected>>, <Return>).
* **Обработка исключений:** Ловит ValueError (некорректное число), ZeroDivisionError и другие.
* **Базовая единица:** Для большинства категорий конвертация идет через базовую единицу (e.g., метры для длины).
* **Особенности:** Температура обрабатывается отдельно из-за нелинейных формул. Для объема реализована специальная логика (см. ошибки ниже).

**3. Схемы и диаграммы**

**Диаграмма классов (UML-в текстовом формате)**

|  |
| --- |
| UnitConverter |
| Атрибуты:  - root: Tk  - units: dict {категория: список единиц}  - conversion\_factors: dict {категория: коэффициенты}  - category\_var: StringVar  - from\_var: StringVar  - to\_var: StringVar  - input\_var: StringVar  - result\_var: StringVar  - result\_info\_var: StringVar |
| Методы:  + \_\_init\_\_(root): Инициализация окна и UI  + setup\_ui(): Создание интерфейса  + update\_units(event): Обновление единиц  + swap\_units(): Обмен единиц "Из" и "В"  + clear\_fields(): Очистка полей ввода/вывода  + convert\_temperature(value, from, to): Конвертация температуры  + convert(): Выполнение конвертации |

**Схема интерфейса (ASCII-арт)**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 - Интерфейс

**Диаграмма потока данных для конвертации**

1. Пользователь выбирает категорию → update\_units обновляет комбобоксы.
2. Ввод значения + единицы → Нажатие "Конвертировать" → convert:
   * Проверки (значение, категория, единицы).
   * Если Температура: convert\_temperature.
   * Иначе: value \* factor\_from / factor\_to.
   * Форматирование → Отображение в result\_var и result\_info\_var.