

Техническое задание проекта "Решатор задач"

1. Логика проекта

2. Функциональные требования

1. Экран 1: Выбор фигуры

- Пользователь выбирает геометрическую фигуру, для которой он хочет выполнить расчёт.
- Возможные действия:
- **Длина окружности** — выбор этой опции переводит пользователя на экран ввода данных, необходимых для расчёта длины окружности.
- **Найти площадь** — выбор этой опции переводит на экран ввода данных для расчёта площади выбранной фигуры.
- Интерактивные элементы:
- Выбор типа фигуры (кнопки, переключатели и т.д.).

1. Экран 2: Ввод известных данных

- После выбора фигуры пользователь вводит известные параметры (например, радиус, диаметр, сторону), в зависимости от типа фигуры и требуемого расчёта.
- Поля ввода данных:
- Поля для ввода значений параметров, необходимых для расчёта (например, радиус или диаметр для окружности, сторона для квадрата).
- Логика переходов:
- Переход к экрану расчёта после ввода данных.

1. Экран 3: Вывод результата

- Экран отображает результат расчёта.
- **Отдельное окно** — используется для вывода дополнительных данных или полного решения в модальном окне.
- **Вывод решения текстом** — отображение результата в текстовом формате.
- Интерактивные элементы:
- Кнопка для возврата к экрану выбора фигуры или ввода данных для нового расчёта.

3. Дополнительные функции

- **Строение фигуры по известным данным** — визуальное отображение фигуры на экране с заданными параметрами.
- **Разработка универсального алгоритма расчета** — логика, позволяющая адаптировать расчёт для различных типов фигур и параметров.

4. Интерфейсные требования

- Интерфейс должен быть минималистичным и интуитивным.
- Элементы управления (кнопки, поля ввода, выпадающие списки) должны быть чётко видимы и подписаны.

- Поле вывода результата должно быть с крупным шрифтом для удобства восприятия.
- Должна быть возможность вернуться к предыдущим этапам (например, выбор фигуры) без потери введённых данных.

5. Требования к производительности

- Программа должна выполнять расчёты мгновенно и не задерживать переходы между экранами.
- В случае сложных расчётов результат должен отображаться не более чем через 1 секунду после ввода данных.

6. Требования к надёжности

- Программа должна обрабатывать ошибки ввода данных (например, предупреждать о неверно введённых данных или отсутствующих параметрах).
- Предусмотреть защиту от ошибок, связанных с несанкционированными изменениями кода.

7. Требования к тестированию

- Тестирование ввода данных для всех поддерживаемых фигур и расчётов.
- Проверка корректности вывода результатов в разных ситуациях и для различных объемов данных.
- Проверка быстродействия и функционала каждого экрана на соответствие требованиям.

8. Документация

- **Руководство пользователя** с описанием работы с программой и её функционала.
- **Техническая документация** с описанием алгоритмов расчёта и структуры кода программы.

Виджеты и библиотеки

Основные виджеты:

1. **QWidget** — базовый класс для всех виджетов.
2. **QMainWindow** — основной виджет окна приложения.
3. **QDialog** — диалоговые окна для ввода данных или подтверждений.

Виджеты для пользовательского интерфейса:

1. **QLabel** — для отображения текста и изображений.
2. **QLineEdit** — однострочное поле для ввода текста (например, условий задачи).
3. **QTextEdit** — многострочное поле для ввода текста (для сложных задач или пояснений).

4. **QComboBox** — выпадающий список для выбора из предустановленных опций (например, выбор фигуры).
5. **QPushButton** — кнопки для выполнения действий.
6. **QCheckBox** — флажки для включения дополнительных параметров.
7. **QRadioButton** — радиокнопки для выбора одного из вариантов.
8. **QSlider** — ползунок для выбора числовых значений (например, размера фигур).
9. **QSpinBox** — поле для ввода чисел с возможностью изменения значений.
10. **QProgressBar** — индикатор выполнения процесса.

Виджеты для отображения графики:

1. **QGraphicsView** — виджет для отображения графики и фигур.
2. **QGraphicsScene** — сцена для управления графическими элементами.

Контейнеры и компоновщики:

1. **QVBoxLayout** — вертикальный компоновщик для размещения виджетов в столбик.
2. **QHBoxLayout** — горизонтальный компоновщик для размещения виджетов в строку.
3. **QGridLayout** — сеточный компоновщик для размещения виджетов в виде таблицы.
4. **QStackedWidget** — контейнер для переключения между несколькими виджетами.

Дополнительные виджеты:

1. **QTabWidget** — вкладки для организации разделов приложения.
2. **QToolBar** — панель инструментов с кнопками и действиями.
3. **QStatusBar** — строка состояния внизу окна для отображения сообщений.

Диалоговые окна:

1. **QFileDialog** — окно для выбора файлов (например, для сохранения решений).
2. **QMessageBox** — окно для отображения сообщений или предупреждений.

Дизайн

- **Стиль:** Современный и минималистичный, акцент на функциональность.
- **Цветовая палитра:** Спокойные тона (например, синий, зелёный, белый) для удобного восприятия.
- **Шрифты:** Читаемые шрифты без засечек для основного текста и заголовков.
- **Иконки:** Простые и понятные, соответствующие действиям.