

# # Техническое задание на разработку консольного приложения архиватора

## ## 1. Введение

Цель данного документа - описать требования и спецификации для разработки консольного приложения архиватора, предназначенного для архивации (сжатия) и распаковки файлов и директорий.

## ## 2. Общие сведения

Название приложения: Archiver.

Платформа: Консольные приложения под операционные системы Windows, Linux и macOS.

## ## 3. Функциональные требования

### ### 3.1. Основные возможности

#### 1. Архивация файла(ов) или директории:

- Пользователь должен иметь возможность выбрать файлы или директории для архивации.
- Приложение должно поддерживать архивацию в форматы: ZIP, TAR.
- Должна быть возможность указать имя выходного архивного файла.
- Предусмотреть возможность выбора степени сжатия.

#### 2. Распаковка архива:

- Поддержка распаковки архивов из форматов ZIP и TAR.
- Возможность указать директорию для распакованных файлов.

#### 3. Прочие функции:

- Просмотр содержимого архива без его распаковки.
- Добавление файлов в существующий архив.
- Удаление файлов из архива.
- Возможность тестирования архива на ошибки с выводом соответствующих ошибок

### ### 3.2. Параметры запуска

Приложение должно поддерживать запуск с параметрами из командной строки, указывая необходимые операции и опции. Примеры параметров запуска:

- **-c** или **--create** – создать архив.
- **-x** или **--extract** – извлечь архив.
- **-l** или **--list** – просмотреть содержимое архива.
- **-a** или **--add** – добавить файлы в архив.
- **-d** или **--delete** – удалить файлы из архива.
- **-f** или **--file** – указать путь к архиву или к файлам для архивации.
- **-o** или **--output** – путь для сохранения архива или извлеченных файлов.
- **-v** или **--verbose** – режим с детальным выводом процесса работы.
- **-s** или **--size** – указание уровня сжатия (только для архивации).

### ### 3.3. Взаимодействие с пользователем

Пользовательское взаимодействие с приложением осуществляется через командную строку. Ошибки выполнения команд и инструкции по использованию опций должны корректно отображаться пользователю.

## ## 4. Нефункциональные требования

### ### 4.1. Производительность

- Время архивации/распаковки не должно превышать времени, затраченного стандартными средствами ОС для аналогичного объема данных более чем на 20%.

### ### 4.2. Портативность

- Приложение должно быть исполняемым без установки дополнительных библиотек или средств разработки на Windows, Linux и macOS.

### ### 4.3. Удобство использования

- Предусмотреть понятные и четкие сообщения об ошибках и подсказки по использованию приложения.
- Поддержка справки по командам приложения через аргумент **--help** или **-h**.

## ## 5. Тестирование

### ### 5.1. Тестирование функционала

- Тестирование всех ключевых функций: архивация, распаковка, просмотр содержимого, добавление и удаление файлов.
- Проверка работы в различных операционных системах.
- Тестирование с различными размерами и типами файлов.

### ### 5.2. Тестирование производительности

- Сравнительный анализ времени выполнения операций с использованием стандартных средств ОС и приложения.

### ### 5.3. Тестирование удобства использования

- Проверка наличия и корректности вывода помощи и сообщений об ошибках.

## ## 6. Сроки и этапы разработки

Разработка приложения будет разделена на следующие основные этапы:

1. Планирование – уточнение задач, ресурсов и графиков.
  - a. Добавить преподавателя в свой таск трекер
  - b. Подготовить декомпозицию и планирование задач.
2. Проектирование – разработка архитектуры и дизайна приложения.
3. Реализация – непосредственное написание кода приложения.
4. Тестирование – проверка соответствия функциональным и нефункциональным требованиям.
  - написанные автотесты (юнит и тд.)
5. Деплоймент – подготовка и распространение финальной версии продукта.