ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Канд. техн. наук, доцент департамента программной инженерии факультета компьютерных наук.

_____ Н. С. Белова
«13» мая 2022 г.

УТВЕРЖДАЮ

Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук

_____ В. В. Шилов
«13» мая 2022 г.

ПРОГРАММА ДЛЯ СОЗДАНИЯ КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЙ, ГОТОВЫХ К РАЗВЕРТЫВАНИЮ В КОНТЕЙНЕРНОЙ СРЕДЕ

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.08-01 51 01-1-ЛУ

Исполнитель
студентка группы БПИ197
_____ / Голикова Е. В. /
«13» мая 2022 г.

Москва 2022

Created By: Sign Doc

УТВЕРЖДЕН

RU.17701729.04.08-01 51 01-1-ЛУ

ПРОГРАММА ДЛЯ СОЗДАНИЯ КЛИЕНТ-СЕРВЕРНЫХ ПРИЛОЖЕНИЙ, ГОТОВЫХ К РАЗВЕРТЫВАНИЮ В КОНТЕЙНЕРНОЙ СРЕДЕ

Программа и методика испытаний

RU.17701729.04.08-01 51 01-1

Листов 22

Москва 2022

Содержание

1. Объект испытаний 3
1.1. Наименование программы 3
1.2. Область применения 3
2. Цель испытаний 4
3. Требования к программе 5
3.1. Требования к функциональным характеристикам 5
3.2. Требования к интерфейсу 5
3.3. Требования к надежности 5
4. Требования к программной документации 8
4.1. Предварительный состав программной документации 8
4.2. Специальные требования к программной документации 8
5. Средства и порядок испытаний 10
5.1. Технические и программные средства 10
5.2. Порядок проведения испытаний 10
6. Выполнение программы 11
6.1. Установка и запуск приложения 11
6.2. Испытание выполнения требований к программной документации 11
6.3. Испытание выполнения требований к функциональным характеристикам и требований к интерфейсу 11
6.4. Испытания выполнения требований к надежности 17
Список литературы 21

1. Объект испытаний

1.1. Наименование программы

Наименование программы – «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде» («Program for Creating Client-Server Applications, Ready for Deployment in the Container Environment»).

1.2. Область применения

Приложение предназначено для использования программистами, в частности – fullstack-разработчиками, начинающими программистами, IT-консультантами компаний и другими специалистами из IT-сферы, которым нужно быстро и легко автоматизировать существующую систему или же создать с нуля каркас для нового проекта, который можно быстро доработать и пустить в использование.

Приложение представляет из себя программу, которая по характеристикам, заданным декларативно пользователем, позволяет создать базу данных, серверную и/или клиентскую части, а также подготовить эти компоненты к развертыванию в контейнерной среде.

2. Цель испытаний

Целью проведения испытаний является проверка на соответствие требованиям, изложенным в прилагаемом программном документе «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде. Техническое задание» из комплекта документации в соответствии с требованиями ГОСТ.

3. Требования к программе

3.1. Требования к функциональным характеристикам

Программа должна обеспечивать выполнение следующих функций:

1) Принимать на вход в качестве параметров командной строки путь к файлу с характеристиками компонентов и путь к директории, в которой будут располагаться результирующие файлы и директории.
2) Создать, если требуется, скрипты для создания базы данных с использованием СУБД PostgreSQL [1] и для ее заполнения таблицами согласно заданным характеристикам и разместить их в выходной директории.
3) Создать, если требуется, проект с серверной частью с использованием фреймворка Spring [2] на языке Java [3] согласно заданным характеристикам и разместить директорию проекта в выходной директории.
4) Создать, если требуется, проект с клиентской частью в виде Android-приложения [4] согласно заданным характеристикам и разместить директорию проекта в выходной директории.
5) Реализовать, если требуется, возможность для запуска контейнеров требуемых компонентов в сети.

3.2. Требования к интерфейсу

Программа должна представлять из себя консольное приложение, запускаемое вызовом его в командной строке с параметрами, позволяющее пользователю использовать функции, описанные выше.

3.3. Требования к надежности

При любом действии пользователя программа не должна завершаться аварийно. При неправильном формате вводимых данных программа должна выводить сообщение с предупреждением о неправильном формате данных. При возникновении исключительной ситуации программа должна вывести сообщение об обработанной ошибке (см. Таблица 1) и принудительно завершить создание компонента.

Таблица 1. Описание сообщений оператору

Описание ошибки	Текст ошибки	Реакция пользователя

Ошибка во время валидации команды – некорректное значение опции in-path	Failed to validate <значение опции>: Must be correct absolute path to existing .yml/.yaml-file with quotes.	Ввести команду заново, исправив путь к файлу на корректный.
Ошибка во время валидации команды – некорректное значение опции out-path	Failed to validate <значение опции>: Must be correct absolute path to existing directory with quotes.	Ввести команду заново, исправив путь к директории на корректный.
Введена команда без опции out-path	Required option --out-path must be specified.	Ввести команду заново, указав все обязательные опции.
Введена команда без опции in-path	Required option --in-path must be specified.	Ввести команду заново, указав все обязательные опции.
Синтаксис YAML-файла некорректен	Something went wrong. Errors occurred while validating input file. Please fix them and try again. Error message: cannot scan file contents. Check correctness of YAML syntax and absence of backslashes, fix and try again.	Ввести команду заново, исправив синтаксис файла на корректный.
Структура файла не соответствует требуемой или какие-либо значения некорректны	Something went wrong. Errors occurred while validating input file. Please fix them and try again. Path: <путь к ошибке> Error message: <сообщение библиотеки об ошибке>	Ввести команду заново, исправив формат файла на корректный.
Произошла любая другая непредвиденная	Something went wrong. Errors occurred while validating input file. Please fix them and try again.	Попробовать запустить программу еще раз, либо

ситуация при парсинге и валидации	Error: unexpected error. Contact us to solve issue.	связаться с разработчиками.
Произошла ошибка во время создания скриптов для создания базы данных	Something went wrong while creating sql-scripts. Try again or contact us to solve issue.	Попробовать запустить программу еще раз, либо связаться с разработчиками.
Произошла ошибка во время создания проекта сервера	Something went wrong. Maybe, you do not have Spring or JDK installed.	Проверить наличие необходимых компонентов на компьютере.
Произошла ошибка во время заполнения проекта сервера	Something went wrong while filling project. Maybe, there are troubles with file paths. Try again or contact us to solve issue.	Попробовать запустить программу еще раз, либо связаться с разработчиками.
Произошла ошибка во время создания проекта клиента	Something went wrong. Maybe, you do not have Android SDK or Gradle or JDK installed.	Проверить наличие необходимых компонентов на компьютере.
Произошла ошибка во время заполнения проекта клиента	Something went wrong while filling project. Maybe, there are troubles with file paths. Try again or contact us to solve issue.	Попробовать запустить программу еще раз, либо связаться с разработчиками.
Произошла ошибка по время создания скрипта для контейнеризации	Something went wrong while creating containerization scripts. Try again or contact us to solve issue.	Попробовать запустить программу еще раз, либо связаться с разработчиками.
Произошла любая другая ошибка	Something went wrong. Exception message: <сообщение об ошибке>.	Попробовать запустить программу еще раз, либо связаться с разработчиками.

4. Требования к программной документации

4.1. Предварительный состав программной документации

В рамках данной работы должна быть разработана следующая программная документация в соответствии и ГОСТ ЕСПД:

- «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде». Техническое задание [5];
- «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде». Программа и методика испытаний [6];
- «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде». Текст программы [7];
- «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде». Пояснительная записка [8];
- «Программа для создания клиент-серверных приложений, готовых к развертыванию в контейнерной среде». Руководство оператора [9];

4.2. Специальные требования к программной документации

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.);

Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ» за 7 дней до начала периода защит.

Документация и программа сдаются в электронном виде в формате .pdf или .docx в архиве формата .zip или .rar.

За три дня до начала периода защит все материалы курсового проекта:

- комплект программной документации,
- отчет из системы «Антиплагиат»,
- коды или ссылка на репозиторий,
- документы по NDA при необходимости,
- отзыв руководителя,
- другие необходимые материалы

должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект, 3 курс ПИ» в личном кабинете в информационной образовательной среде Smart LMS (Smart Learning Management System) НИУ ВШЭ.

Презентация доклада должна быть загружена не позднее, чем за один день до начала периода защит.

5. Средства и порядок испытаний

5.1. Технические и программные средства

Во время проведения испытаний использовалась система со следующими характеристиками:

1) 8 Гб ОЗУ.
2) Операционная система Windows 11.
3) Установленный JDK 11 [10].
4) Установленный Android SDK.
5) Установленный Gradle [11].
6) Установленный Spring Boot.
7) Установленный Spring Boot CLI [12].
8) Свободный доступ к сети Интернет.

5.2. Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

1) Проверка требований к программной документации.
2) Проверка требований к интерфейсу.
3) Проверка требований к составу выполняемых функций.
4) Проверка требований к надежности.

6. Выполнение программы

6.1. Установка и запуск приложения

Для дальнейших испытаний необходимо перейти по ссылке https://disk.yandex.ru/d/4WsJOVPUoFGpGQ, где опубликован файл приложения. Для установки необходимо загрузить jar-файл, расположенный по этой ссылке, к себе на устройство, а затем для дальнейших действий нужно открыть стандартную командную строку.

6.2. Испытание выполнения требований к программной документации

Состав программной документации проверяется визуально, проверяется наличие всех подписей и наличие программной документации в системе Smart LMS. Также визуально проверяется соответствие документации требованиям ГОСТ. Все документы удовлетворяют представленным требованиям.

6.3. Испытание выполнения требований к функциональным характеристикам и требований к интерфейсу

Пользователь должен иметь возможность запустить загруженный jar-файл у себя в командной строке путем ввода команды вида:

java -jar <путь к jar-файлу> start <опции>

Сразу же после запуска, при условии корректного ввода и отсутствия непредвиденных ошибок, программа должна начать свое выполнение и уже вскоре выдать результат, а затем завершить работу.

Среди опций, которые должны быть доступны, должны быть опции для передачи путей к входному файлу и выходной директории.

Чтобы дать программе команду на создание необходимых компонентов, пользователь должен иметь возможность использовать опцию --in-path (или ее краткую форму -i) и опцию --out-path (или ее краткую форму -o). Для каждой опции должна быть возможность указать ей значение.

Таким образом, пользователь должен иметь возможность ввести следующие команды:

java -jar <путь к jar-файлу> start --in-path <значение> --out-path <значение>

или

java -jar <путь к jar-файлу> start --out-path <значение> --in-path <значение>

или

java -jar <путь к jar-файлу> start -i <значение> -o <значение>

или

java -jar <путь к jar-файлу> start -o <значение> -i <значение>

Значением опции --out-path должен являться корректный путь к существующей директории на компьютере пользователя.

Значением опции --in-path должен являться корректный путь к существующему yaml- или yml- файлу, содержание которого должно представлять из себя декларативно описанные характеристики компонентов, которые нужно создать, оформленные в виде YAML-разметки [13] с корректной структурой и синтаксисом. При этом содержание файла для испытаний штатного поведения программы должно соответствовать заданному формату:

db: # спецификации для модуля создания базы данных, опционально

postgresql: # желаемая СУБД, в данной версии – только PostgreSQL

db-name: <имя базы данных – любое>

host: <корректный ip-адрес компьютера, на котором будет запускаться PostgreSQL-сервер>

username: <корректное имя пользователя PostgreSQL>

password: <пароль для указанного пользователя>

tables: # список таблиц для создания; допустимо любое количество таблиц

- table-name: <имя таблицы - любое>

columns: # список столбцов таблицы; не более 1600 столбцов

- col-name: <имя колонки - любое>

opts: <тип данных; number, string, bool или date>

- …

- …

server: # спецификации для модуля создания сервера, опционально

spring: # желаемый фреймворк, в данной версии – только Spring

project: # спецификации для создания проекта

build: <система сборки; опционально; maven или gradle; по умолчанию - gradle>

language: <язык проекта; опционально; в данной версии – только java>

boot-version: <любая существующая версия Spring Boot; опционально; по умолчанию – 2.6.2>

group: <группа проекта; опционально; по умолчанию - com.example>

artifact: <название артефакта; опционально; по умолчанию - demo>

proj-name: <название проекта; опционально; по умолчанию - demo>

description: <описание проекта; опционально; по умолчанию - "Demo project for Spring Boot">

packaging: <тип архива; опционально; jar, war, pom, ear, rar или par; по умолчанию - jar>

java-version: <версия Java; опционально; любая с 8 по 11; по умолчанию - 11>

project-version: <версия проекта; опционально; по умолчанию - 0.0.1-SNAPSHOT>

deps: <список зависимостей для проекта; опционально; полный список доступынх зависимостей можно получить с помощью команды spring init --list>

- <зависимость 1>
- <зависимость 2>
- …

properties: # свойства проекта

db: # информация о базе данных, с которой будет соединяться сервер

type: <СУБД, любая>

username: <корректное имя пользователя>

password: <пароль для указанного пользователя>

sql-host: <корректный ip-адрес базы данных>

sql-port: <корректный порт базы данных>

db-name: <название базы данных>

controllers: # список хендлеров – контроллеров, допустимо любое количество

- **controller-name:** <название контроллера в формате <имя сущности>Controller>

requests: # список запросов для данного контроллера, допустимо любое количество

- **req-name:** <имя метода запроса, в корректном для Java-метода формате>

uri: <адрес запроса, корректный URI>

mapping: <тип запроса; post, put, get, patch или delete>

- **…**

- **…**

client: # спецификации для модуля создания клиента, опционально

android: # тип проекта, в данной версии – только Android-приложение

proj-name: <имя проекта, любое допустимое для java-приложения; по умолчанию - demo>

language: <язык проекта, в данной версии только java>

package-name: <имя корневого пакета, должно содержать хотя бы одну точку; по умолчанию – demo.example>

test-framework: <фреймворк для тестирования; junit, testng, spock, junit-jupiter; по умолчанию – junit-jupiter>

server-host: <корректный ip-адрес сервера, с которым взаимодействует клиент>

server-port: <корректный порт сервера, с которым взаимодействует клиент>

requests: # http-запросы, допустимо любое количество

- **req-name:** <имя метода запроса, в корректном для java-метода формате>

  uri: <адрес запроса, корректный URI>

  type: <тип запроса; post, put, get или delete>

  entity: <имя сущности, с которой оперирует запрос, в корректном для java-класса формате>

- **…**

containerization: # спецификации для модуля контейнеризации, опционально

docker: # используемая платформа, в данной версии – только Docker

jars: # спецификации для jar-файлов, для которых требуется создать контейнеры; опционально; допустимо любое количество

- **image-name:** <название будущего Docker-образа, любое допустимое>

  container-name: <название будущего Docker-контейнера, любое допустимое>

  dir-name: <название директории, где будет собираться Docker-образ, любое допустимое для Windows-директории>

  jar-path: <пусть к существующему jar-файлу>

- **…**

nginx: # спецификации для контейнеров NGINX-веб-серверов; опционально; допустимо любое количество

- **image-name:** <название будущего Docker-образа, любое допустимое>

  container-name: <название будущего Docker-контейнера, любое допустимое>

  dir-name: <название директории, где будет собираться Docker-образ, любое допустимое для Windows-директории>

  backend-container-name: <имя контейнера связанного сервера>

- **…**

postgres: # спецификации для контейнеров PostgreSQL-баз данных; опционально; допустимо любое количество

- **container-name:** <название будущего Docker-контейнера, любое допустимое>

  password: <пароль пользователя PostgreSQL>

- **…**

network: # спецификации сети, в которой будут работать контейнеры

  network-name: <имя сети, любое допустимое>

Все поля, не помеченные как опциональные, должны являться обязательными.

Создание каждого компонента должно являться опциональным, но при этом хотя бы один (любой) компонент должен присутствовать в спецификации, иначе валидация файла не должна проходить.

При штатной отработке программы в консоли должны появляться следующие сообщения в зависимости от происходящего (рис. 1):

Сообщение	Значение
Parsing input file...	Происходит процесс обработки входного файла со спецификациями
Parsed, no errors found!	Обработка прошла успешно
Creating sql-scripts...	Создаются sql-скрипты для инициализации заполнения базы данных
Sql-scripts created successfully!	Скрипты успешно созданы
Creating server project...	Происходит инициализация проекта сервера
Server project created successfully! Using service at https://start.spring.io Project extracted to <директория размещения проекта>	Проект успешно создан и помещен в указанную папку
Filling created server project...	Происходит заполнение созданного проекта нужными файлами и кодом
Server project successfully filled!	Проект успешно заполнен
Creating android project...	Происходит инициализация клиентского Android-приложения

Client android project created successfully!	Проект успешно создан
Filling created client project...	Происходит заполнение созданного проекта нужными файлами и кодом
Client project successfully filled!	Проект успешно заполнен
Creating containerization scripts...	Происходит создание скриптов для создания образов и запуска контейнеров
Containerization scripts created successfully!	Скрипты успешно созданы
Finished!	Работа программы завершена

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
Parsed, no errors found!

Creating sql-scripts...
Sql-scripts created successfully!

Creating server project...
Server project created successfully!
Using service at https://start.spring.io
Project extracted to 'C:\Users\Lenovo X1\Documents\ac\result\my_demo_project'

Filling created server project...
Server project successfully filled!

Creating android project...
Client android project created successfully!

Filling created client project...
Client project successfully filled!

Creating containerization scripts...
Containerization scripts created successfully!

Finished!


C:\Users\Lenovo X1>
```

Рисунок 1 - Вывод программы при штатном создании всех четырех компонентов

После окончания работы программы в выходной директории должны появиться все требуемые компоненты (рис. 2).

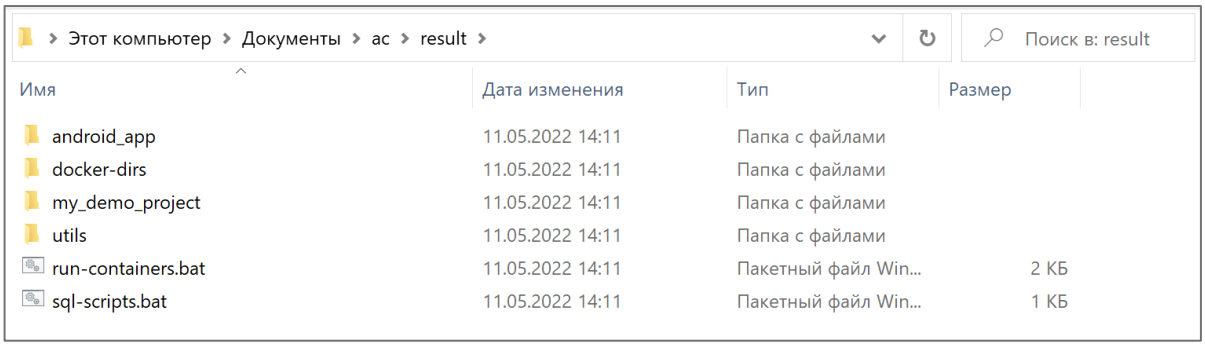

Рисунок 2 - требуемый результат работы программы

Здесь, папки android_app и my_demo_project должны содержать проекты мобильного Android-приложения и сервера на Spring Boot, соответственно, а файла run-containers.bat и sql-scripts.bat должны содержать скрипты для создания образов и контейнеров и скрипты для создания и заполнения базы данных, соответственно.

Как мы можем наблюдать из проведенных испытаний, все описанные требования к функциям и интерфейсу выполняются.

6.4. Испытания выполнения требований к надежности

Программа не должна завершаться аварийно ни при каких действиях пользователя. При неправильном формате вводимых данных программа должна выводить сообщение с предупреждением о неправильном формате данных. При возникновении исключительной ситуации программа должна вывести сообщение об обработанной ошибке согласно Таблице 1 и принудительно завершить создание компонента.

Примеры ошибок, которые должны возникать при валидации входной команды, представлены на рисунках 3-4.

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o "
fake\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
The following errors occurred while parsing your command:

Failed to validate "-o fake\\result": Must be correct absolute path to existing directory
 with quotes.

C:\Users\Lenovo X1>
```

Рисунок 3 – ошибка в случае указания несуществующей выходная директория

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result"
The following errors occurred while parsing your command:

Required option --in-path must be specified.

C:\Users\Lenovo X1>_
```

Рисунок 4 - ошибка в случае отсутствия пути к входному файлу

Тексты ошибок валидации содержания входного файла продуцируются библиотекой валидации malli [14] и могут быть самыми разнообразными, как встроенными, так и кастомными. Образцы некоторых из них представлены на рисунках 5-8.

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
Something went wrong.
Errors occurred while validating input file.
Please fix them and try again.

Path: db >> postgresql >> password
Error message: "missing required key"

Finished!
```

Рисунок 5 - ошибка в случае отсутствия пароля пользователя в компоненте создания базы данных

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o "C:\
Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
Something went wrong.
Errors occurred while validating input file.
Please fix them and try again.

Path: server >> spring >> project >> java-version
Error message: "'20': should be one of: 1.8 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17"


Finished!
```

Рисунок 6 - ошибка в случае указания несуществующей версия Java для проекта сервера

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" sta
rt -o "C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac
\yaml.yaml"
Parsing input file...
Something went wrong.
Errors occurred while validating input file.
Please fix them and try again.

Path: client >> android >> requests >> req-name
Error message: "'Get_user': incorrect method name, must follow java method naming
conventions"

Path: client >> android >> requests >> uri
Error message: "'user': incorrect uri path"

Path: client >> android >> requests >> type
Error message: "'let': should be one of: post | put | get | delete"

Path: client >> android >> requests >> entity
Error message: "'uSeR': incorrect entity name, must start with a capital letter"

Finished!
```

Рисунок 7 - ошибка в случае указания некорректных характеристик http-запроса

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
 "C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml
"
Parsing input file...
Something went wrong.
Errors occurred while validating input file.
Please fix them and try again.

Path: containerization >> docker >> jars >> jar-path
Error message: "'fake.jar': incorrect jar path, must be path to existing jar-file with
slashes, not backslashes"

Finished!
```

Рисунок 8 – ошибка в случае указания несуществующего jar-файла в компоненте контейнеризации

На рисунках 9-10 представлены примеры ошибок, которые должны возникать в том случае, если у пользователя не установлено какое-либо необходимое для работы программы программное обеспечение.

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
Parsed, no errors found!

Creating sql-scripts...
Sql-scripts created successfully!

Creating server project...
Something went wrong. Maybe, you do not have Spring Boot or JDK installed.

Creating containerization scripts...
Containerization scripts created successfully!

Finished!
```

Рисунок 9 - ошибка, возникающая при отсутствии у пользователя установленного Spring

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
Parsed, no errors found!

Creating sql-scripts...
Sql-scripts created successfully!

Creating server project...
Server project created successfully!
Using service at https://start.spring.io
Project extracted to 'C:\Users\Lenovo X1\Documents\ac\result\my_demo_project'

Filling created server project...
Server project successfully filled!

Creating android project...
Something went wrong. Maybe, you do not have Android SDK or Gradle or JDK installed.

Creating containerization scripts...
Containerization scripts created successfully!

Finished!
```

Рисунок 10 - ошибка, возникающая при отсутствии у пользователя установленного Gradle

На рисунке 11 представлен случай «сломанного» YAML-файла, когда синтаксис YAML-разметки не соблюден пользователем.

```
C:\Users\Lenovo X1>java -jar "C:\Users\Lenovo X1\Documents\ac\app-creator.jar" start -o
"C:\Users\Lenovo X1\Documents\ac\result" -i "C:\Users\Lenovo X1\Documents\ac\yaml.yaml"
Parsing input file...
while scanning a simple key
 in 'string', line 3, column 1:
    putting incorrect syntax
    ^
could not find expected ':'
 in 'string', line 4, column 1:
    here
    ^

Something went wrong.
Errors occurred while validating input file.
Please fix them and try again.

Error message: cannot scan file contents. Check correctness of YAML syntax and absence o
f backslashes, fix and try again.
Finished!
```

Рисунок 11 - ошибка, возникающая при использовании файла с некорректным YAML-синтаксисом

Как мы можем наблюдать из проведенных испытаний, все описанные требования к надежности выполняются.

Список литературы

1) Документация PostgreSQL [Электронный ресурс] //URL: https://www.postgresql.org/docs/ (Дата обращения: 06.05.2022, режим доступа: свободный).
2) Spring Boot Overview, [Электронный ресурс] // URL: https://spring.io/projects/spring-boot (Дата обращения: 06.05.2022, режим доступа: свободный).
3) Java Documentation, [Электронный ресурс] //URL: https://docs.oracle.com/en/java/ (Дата обращения: 06.05.2022, режим доступа: свободный).
4) Create an Android Project, [Электронный ресурс] // URL: https://developer.android.com/training/basics/firstapp/creating-project (Дата обращения: 06.05.2022, режим доступа: свободный).
5) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6) ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7) ГОСТ 19.401-78. ЕСПД. Текст программы. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
8) ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
9) ГОСТ 19.505-79. ЕСПД. Руководство оператора. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001
10) JDK 11 - OpenJDK [Электронный ресурс] //URL: https://openjdk.java.net/projects/jdk/11/ (Дата обращения: 06.05.2022, режим доступа: свободный).
11) Gradle User Manual [Электронный ресурс] //URL: https://docs.gradle.org/current/userguide/userguide.html (Дата обращения: 06.05.2022, режим доступа: свободный).
12) Spring Boot CLI [Электронный ресурс] //URL: https://docs.spring.io/spring-boot/docs/current/reference/html/cli.html (Дата обращения: 06.05.2022, режим доступа: свободный).

13) YAML Syntax – Ansible Documentation [Электронный ресурс] //URL: https://docs.ansible.com/ansible/latest/reference_appendices/YAMLSyntax.html (Дата обращения: 06.05.2022, режим доступа: свободный).
14) metosin/malli: Data-Driven Schemas for Clojure/Script. [Электронный ресурс] //URL: https://github.com/metosin/malli (Дата обращения: 06.05.2022, режим доступа: свободный).