**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Заместитель руководителя департамента  Программной Инженерии  Факультета Компьютерных Наук  профессор  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.М. Гринкруг  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.02.03-01 34 01-1** | | **ИССЛЕДОВАНИЕ И РЕАЛИЗАЦИЯ**  **СЕРИАЛАЙЗЕРА-ДЕСЕРИАЛАЙЗЕРА**  **АЦИКЛИЧЕСКОГО ГРАФА ИЗ**  **ИНСТАНСОВ JAVA BEANS COMPONENT**  **Руководство оператора**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.02.03-01 34 01-1-ЛУ** | | |
|  |  | |
| Исполнитель:  студент группы БПИ171  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Данилов А.А. /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
|  | |  |

**2019**

|  |  |  |  |
| --- | --- | --- | --- |
| **УТВЕРЖДЕНО**  **RU.17701729.02.03-01 34 01-1-ЛУ** | |  |  |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** | **RU.17701729.02.03-01 34 01-1** | | **ИССЛЕДОВАНИЕ И РЕАЛИЗАЦИЯ**  **СЕРИАЛАЙЗЕРА-ДЕСЕРИАЛАЙЗЕРА**  **АЦИКЛИЧЕСКОГО ГРАФА ИЗ**  **ИНСТАНСОВ JAVA BEANS COMPONENT**  **Руководство оператора**  **Руководство оператора**  **RU.17701729.02.03-01 34 01-1**  **Листов 5** | | | | |
|  |  | | | |
|  | | | |
|  | | | | |

**2019**

|  |  |  |
| --- | --- | --- |
|  |  |  |

Оглавление

[1. Назначение программы……………………………………………………………………..3](#_Toc8685387)

[**1.1 Функциональное назначение** 3](#_Toc8685388)

[**1.2 Эксплуатационное назначение** 3](#_Toc8685389)

[2. Условия выполнения программы…………………………………………………………3](#_Toc8685390)

[**2.1 Минимальный состав аппаратных средств** 3](#_Toc8685391)

[Так как проектом является библиотека, то технические средства должны обеспечить работу JVM 3](#_Toc8685392)

[**2.2 Минимальный состав программных средств** 3](#_Toc8685393)

[**2.3 Требования к персоналу (пользователю)** 3](#_Toc8685394)

[3. Выполнение программы……………………………………………………………………4](#_Toc8685395)

[**3.1 Добавление библиотеки в свой проект** 4](#_Toc8685396)

[**3.2 Сериализация объекта** 4](#_Toc8685397)

[**3.3 Десериализция объекта** 4](#_Toc8685398)

[**3.4 Создание собственного синтаксиса для сериалайзера** 4](#_Toc8685399)

[ПРИЛОЖЕНИЕ 1………………………………………………………………………………5](#_Toc8685400)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ……………………………………………………..6](#_Toc8685401)

# **1. Назначение программы**

## **1.1 Функциональное назначение**

В результате сериализации пользователь получает файл, который является сериализованным ациклическим графом составного объекта.

С помощью десериализации можно получить объект класса, который был сериализован в другом месте и, возможно, с использованием другого языка.

В процессе сериализации и десериализации графа, можно запоминать и восстанавливать граф в или из текстовых файлов, записанных в фотматах XML/JSON.

## **1.2 Эксплуатационное назначение**

Конечными пользователями библиотеки могут быть программисты, программные инженеры, а также студенты, желающие сериализовать объекты своей программы.

# **2. Условия выполнения программы**

## **2.1 Минимальный состав аппаратных средств**

## Так как проектом является библиотека, то технические средства должны обеспечить работу JVM

## **2.2 Минимальный состав программных средств**

1. JDK
2. JVM
3. IDE для java

## **2.3 Требования к персоналу (пользователю)**

Особых требований к пользователю не предъявляется.

# **3.** **Выполнение программы**

В данном разделе описан пример работы с программой.

## **3.1 Добавление библиотеки в свой проект**

Создать проект в любой IDE для java, написать java bean классы и объекты, которые требуется сериализовать, скопировать содержимое папки ”src” находящейся в папке “HSE-2-course-work-java-beans-serializer”

## **3.2 Сериализация объекта**

Для сериализации необходимо созздать инстанс класса «JavaSerializer» и передать в конструкторе параметром объект который хотим сериализовать.

Приммер:

JavaSerializer serializer = new JavaSerializer(c1);

Далее для начала сериализации следует вызвать метод make класса JavaSerializer с первым параметром FileOutputStream и вторым параметром класс, реализующий интерфейс «SerializingSyntacse»

Пример:

FileOutputStream fileWriter = new FileOutputStream("test5.xml");

serializer.Make(fileWriter, new XmlSerializingSyntacse());

## **3.3 Десериализция объекта**

Для десериализации необходимо создать инстанс класса «JavaDeserializer»

Пример:

JavaDeserializer javaDeserializer = new JavaDeserializer();

Далее для запуска десериализации следует вызвать метод make класса JavaDeserializer с первым параметром String – путь к сериализованному файлу, а вторым параметром тип десериализации. Объект воззвращенный методом make является результатом десериализцаии

Пример:

JavaDeserializer javaDeserializer = new JavaDeserializer();

CyclicGraph pe = (CyclicGraph) javaDeserializer.Make("test5.xml", new XmlDeserializing());

## **3.4** **Создание собственного синтаксиса для сериалайзера**

Для создания собственного синтаксиса для сериалайзера необходимо создать класс, имплементирующий интерфейс «SerializingSyntacse». Для этого класса написаны JavaDOC, которые упростят понимание каждого метода

# **ПРИЛОЖЕНИЕ** **1**

**ТЕРМИНОЛОГИЯ**

|  |
| --- |
| Таблица 1 |

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| **Сериализация** | Процесс преобразования структур данных или состояния объекта в формат, который может быть сохранен (например, в файле или буфере памяти) |
| **Десериализация** | Обратный процесс сериализации |
| **XML** | Расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). |
| **JSON** | Текстовый формат обмена данными, основанный на JavaScript. Как и многие другие текстовые форматы, JSON легко читается людьми. |

# **ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированх |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |