|  |
| --- |
| МИНОБРНАУКИ РОССИИ  Федеральное государственное  бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий

Кафедра корпоративных информационных систем

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **по лабораторной работе №4** | | |
| **по дисциплине** | | |
| **«Объектно-ориентированное программирование»**  **Тема лабораторной работы: «**Ввод и вывод в Java. Пакет java.io и его классы.**»** | | |
| Студент группы | ИКБО-07-18 | Бабух А.А. |
| Принял | ассистент кафедры | Колесников С.Н. |
|  |  |  |
| Выполнено | «\_\_\_» \_\_\_\_\_\_ 201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |  | *(подпись студента)* |
| Зачтено | «\_\_\_» \_\_\_\_\_\_ 201\_\_ г. | *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* |
|  |  | *(подпись преподавателя)* |

* **Постановка задачи**
  1. **Задание №1**

Класс файл.

* + - 1. Управление файловой структурой Java – класс File. Требуется выполнить три упражнения. Результаты каждого упражнения включить в отчет по лабораторной работе. Все упражнения задания реализовать в одном проекте в методе main, разделяя код упражнений комментариями – условие упражнения.
      2. Упражнения по управлению файловой структурой.

Упражнение 1. Исследовать возможности класса File по созданию файлов (пустых) и папок программой. Применение конструктора и метода.

* Создать файл в папке приложения с именем MyFile1.txt.
* Проверить появление файла в папке приложения.
* Создать файл с именем MyFile2.txt в корне определенного диска.
* Проверить появление файла в папке.
* Создать файл с именем MyFile3.txt в папке Имя диска\\Имя папки\\Имя файла.
* Проверить появление файла в папке.
* Создать папку третьего уровня, например, Первая\\Вторая\\Третья.
* Проверить появление папки.
* Все операции заключить в блок try, с обработчиками исключений IOException, Exception, FileNotFoundException.

Упражнение 2. Получить параметры файлов методами класса File. В пунктах задания использовать объекты, созданные в задании 1.

* Проверить, что вызывающий объект содержит имя файла, а не папки и отобразить имя файла, вызывающего объекта и его родительскую папку.
* Проверить, что вызывающий объект содержит имя папки, а не файла и отобразить имя файла, вызывающего объекта.
* Проверить, что в папке приложения существует файл с именем MyFile1.txt.
* Отобразить полный путь к файлу или папке объекта.
* Отобразить размер файла или папки объекта, указать единицу измерения. Прокомментировать вид файла – папка или файл.

Упражнение 3. Модификация файловой структуры приложения средствами класса File.

Добавить в папку приложения еще одну папку.

Сформировать массив файлов, находящихся в папке приложения, используя метод list(). Отобразить содержимое массива.

Сформировать массив файлов, находящихся в папке приложения, используя метод listFiles( ). Отобразить содержимое массива. Определить количество папок, содержащихся в файле приложения.

Удалить папки и файлы созданные во всех трех упражнениях

Задание 2. Использование классов байтовых потоков b потоков DataInputStream и DataOutputStream для записи в файл состояния объектов и файла RandomAccesFile

**1.2 Задание №2**

1. Создать класс, указанный в варианте. Предусмотреть конструктор с параметрами для создания объекта, контролирующий поступающие данные. Данные состояния объекта вводятся с клавиатуры. Дополнительные методы, указанные в варианте
2. Создать класс по управлению файлом, который должен обеспечить следующие функции:
3. Формирование файла данных
   1. Создать выходной поток для записи объекта в файл, располагая файл в папке приложения. Имя файла пользователь вводит с клавиатуры.
   2. Обеспечить проверку существования файла с таким именем и вывод сообщения о его существовании.
   3. Выполнить запись в файл данных нескольких объектам. Данные для полей объекта вводятся с клавиатуры и записываются в файл. Расположите их в файле так, чтобы потом можно было написать алгоритм чтения из файла данных и формирование объектов.. Помните, что при вводе вы будете вводить данные и строковых и примитивных типов (смешанный ввод).
   4. Закрыть выходной поток.
4. Чтение данных
   1. Создать входной поток для чтения данных из созданного файла.
   2. Создать массив с данными о состоянии объектов, используя оду из коллекций Java: для нечетного варианта ArrayList, для четного – LinkedList.
   3. Прочитать данные из файла и разместить в коллекции те объекты, которые, удовлетворяют критериям дополнительного задания варианта (задание 1).
5. Используя класс RandomAccessFile произвольного доступа к записям файла обеспечить:
   1. Создание нового файла из записей фиксированной длины, содержащий сведения по объектам, прочитанных в массив объектов в задании 4.3. Так как поток RandomAccessFile в качестве источника может иметь только массивы примитивного типа, необходимо предусмотреть структуру записи – строка и ее размер. Для этого расширяйте строковые значения пробелами справа, так, чтобы в сумме с примитивными значениями получить нужный размер. При таком условии метод seek можно использовать для перемещения по записям.
   2. Выполнить над новым файлом операцию задания 2 из дополнительного задания варианта.
   3. **Задание №3**

Упражнение 1. Чтение из одного файла текстового файла и запись в другой.

* Создайте два текстовых файл средствами среды или другого текстового редактора: T1.txt и T2.txt. В файл T1.txt запишите несколько символов. Файл T2.txt не заполняйте.
* Определите один поток для чтения из файла T1.txt, а второй выходной с добавлением символов в конец.
* Прочитайте из файла T1.txt значение методом read и перепишите его в файл T2.txt методом write. Перепишите все значения из файла T1.txt в T2.txt. Для выполнения операции создайте цикл, пока метод read не вернет -1.
* При разработке кода используйте исключения на корректность операций ввода и вывода IOException.
* Выполните отладку программы упражнения. Проверьте заполнение файла T2.txt.

Упражнение 2. Применение буферизированных потоков для чтения и записи текстовых файлов. Используйте буфер в 128 байтов.

* Создайте текстовый файл A.txt и запишите в него 512 символов.
* Определите поток inb класса BufferReader и свяжите его с созданным файлом, установив размер буфера равным 128 байтам (файл должен быть прочитан за четыре раза).
* Создайте пустой текстовый файл B.txt
* Создайте поток outb класса BufferWriter и свяжите его с файлом B.txt установив размер буфера 128 байтов.
* Определите буферную переменную buf размером 128 символов – массив из 128 символов.
* Перепишите все данные из файла A.txt в файл B.txt, используя буфер и методы: read(buf) и write(buf). Располагая каждый вывод буфера на отдельной строке файла.

Упражнение 3. Настройка кодировки символов для входного и выходного потоков. Создайте файл с именем A.txt, используя возможности среды (щелкните по csr правой кнопкой и в контекстном меню выполните newFile введите имя). Подготовьте в файле A.txt предложение на русском языке. Вы подготовили текст в кодировке UTF-8.

* 1. **Задание №4**

Задание 4. Применение классов ObjectOutputStream и ObjectInputStream для сериализации и десериализации объектов

4.1.1 Требования по выполнению задания Требуется разработать три класса:

* Управление тестовым файлом. ClassTextFile - исходный файл с данными для объектов.
* Класс информационного объекта на основании одного класса из модели варианта лабораторной работы 2 – имя задайте самостоятельно.
* Управление файлами с сериализованными объектами - ClassSeriyazableFile. Разработать приложение, демонстрирующее выполнение операций над файлом с применением потоков объектов.

4.1.2 Детализация требований по выполнению задания 1. В модели лабораторной работы 2 выберите один класс, объекты которого требуют больше данных. Создайте текстовый файл с данными для заполнения нескольких объектов. Данные разного типа лучше располагать на отдельных строках файла. Разработайте класс ClassTextFile для управления текстовым файлом исходных данных. Поля – путь к файлу. Методы:

* Конструктор с параметром – путь к файлу, с проверкой существования файла.
* Заполнить объект данными файла и вернуть его. Для этого использовать поток Scanner для считывания из текстового файла данных для заполнения полей объекта. Метод считывания должен соответствовать типу поля объекта. Разработать класс для представления объекта, вашего варианта лабораторной работы 2. Поля класса:
* свойства выбранного объекта.

Методы:

* Ввод данных для объекта с консоли.
* Вернуть объект.
* Залить данные в объект.

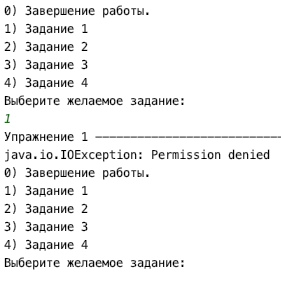
Разработать класс, обеспечивающий запись в файл и чтение из файла, используя классы FileInputStream, FileOutputStream, и классы, обеспечивающие сериализацию объектов ObjectOutputStream (сериализация объектов) и ObjectInputStream (десериализация объектов).

1. **Описание используемых классов**

* Main.java: находится интерфейс программы;
* СlassForObject: ввод данных объектов;
* ClassFromTwoLaba: класс, взятый из второй лабы;
* ClassSeriyazableFile: для дополнения данных из второй лабы;
* ClassTextFile: для ввода данных в файл из второй лабы;
* Sportsman: для списка параметров спортсменов;
* Task2: для ввода данных для спортсменов;

1. **Пользовательский интерфейс**

Пользователь имеет возможность выбрать любое из четырех заданий, а затем любые подзадания. (Смотреть Рисунок №1)



**Рисунок 1 –** Пользовательский интерфейс