

**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Уральский государственный университет имени первого президента России  
Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий  
Центр ускоренного обучения

## **Отчёт по лабораторной работе №10**

**«Система ввода/вывода в Java. Работа с файлами через байтовые  
потоки»**

Руководитель ст. преподаватель

Студент гр. РИЗ-100028у

Н. А. Архипов

И. С. Арсентьев

**Екатеринбург 2021**

## Лабораторная работа №10.

### «Система ввода/вывода в Java. Работа с файлами через байтовые потоки»

*Цель:* получение навыков работы с каталогами и файлами операционной системы, а также с классами ввода/вывода, получение навыков ввода/вывода данных файла через символьные потоки.

*Описание задачи:*

В ходе лабораторной работы составить 10 программ по представленным задачам, представить листинги программ, во вложении к отчёту приложить файлы готовых классов.

Ход выполнения задач:

#### ***1. 1. Создание файлов и папок.***

```
/*Создание файлов и папок.*/
import java.io.File;
public class klassFile1 {
    public static void main(String[] args) {
        try {
            // Создание файла в текущей папке (где расположен файл
klassFile1.java)
            File f1=new File("MyFile1.txt");
            f1.createNewFile();
            if (f1.exists()) {
                //имя нового файла
                System.out.println("Создан!!!!");
                System.out.println("Полный путь1: "+
f1.getAbsolutePath());
            }
            // Создание файла на диске D и вывод полного пути
            File f2=new File("D:\\Test\\MyFile2.txt");
            f2.createNewFile();
            System.out.println( "Полный путь 2: "+
f2.getAbsolutePath());
            // Создание нескольких вложенных папок
            File f3=new File("D:\\Test\\Папка1\\Папка2\\Папка3");
            f3.mkdirs();
            System.out.println("Полный путь 3: "+
```



```

        if(oneByte != -1)
        {
            res += (char)oneByte;
        }
        else
        {
            res += "\n end";
            break;
        }
    }
    return res;
}
}
} //конец программы

```

### ***3. Прочитать и вывести на экран информацию из предварительно созданного файла с использованием буфера в 5 байт..***

```

/*Прочитать и вывести на экран информацию из предварительно
созданного файла с использованием буфера в 5 байт. */
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Arrays;
public class KlassFile3 {
    public static void main(String[] args) {
        FileInputStream fileInputStream = null;
        try {
            String filePath = "D:\\Test\\MyFile2.txt";
            fileInputStream = new FileInputStream(filePath);

            System.out.println(ReadBytesFromStream(fileInputStream) +
"\n\n\n");

        } catch (Exception e) {
            System.out.println("Произошла ошибка " +
e.getMessage());
        }
        finally {
            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (Exception e2) {

```

```

        /*NOP*/
    }
}
}
}
    public static String ReadBytesFromStream(InputStream ins)
throws IOException
    {
        int BUFF_SIZE = 5;
        byte[] buff = new byte[BUFF_SIZE];
        var res = "";
        while(true)
        {
            int byteCount = ins.read(buff);
            if(byteCount != -1) {
                var outputFormat = "количество = %d\nbuff
= %s\nstr=%s\n";
                res = res + String.format(outputFormat, byteCount,
Arrays.toString(buff), new String(buff, 0, byteCount, "cp1251"));
            }
            else {
                break;
            }
        }
        return res;
    }
}
} //конец программы

```

**4. Создать первый файл на диске и записать в него заданное количество вещественных чисел. Далее создать второй файл и переписать в него все числа из первого файла.**

```

/*Создать первый файл на диске и записать в него заданное
количество вещественных чисел. Далее создать второй файл и
переписать в него все числа из первого файла. */
import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.EOFException;
import java.io.File;

```

```

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
public class KlassFile4 {
    public static void main(String[] args) {
        FileOutputStream fos = null;
        FileInputStream fis = null;
        try {
            fos = new
FileOutputStream(GetFileByName("numIsh.txt"));
            FillFile(fos);
            fos.close();
            fis = new
FileInputStream(GetFileByName("numIsh.txt"));
            fos = new
FileOutputStream(GetFileByName("numRez.txt"));
            CopyFile(fis, fos);
            fos.close();
            fis.close();
        } catch (Exception e) {
            System.out.println("Произошла ошибка " +
e.getMessage());
        }
        finally {
            if (fos != null) {
                try {
                    fos.close();
                } catch (Exception e2) {
                    /*NOP*/
                }
            }
            if (fis != null) {
                try {
                    fos.close();
                } catch (Exception e2) {
                    /*NOP*/
                }
            }
        }
        public static File GetFileByName(String filename) throws
IOException {
            String dirPath = "C:\\Test\\%s";

```

```

        File f1=new File(String.format(dirPath, filename));
        f1.createNewFile();
        return f1; }
    public static void FillFile(FileOutputStream fs) throws
IOException
    {
        System.out.println("Введите числа\n");
        BufferedReader ds = new BufferedReader(new
InputStreamReader(System.in));
        DataOutputStream dos = new DataOutputStream(fs);
        try {
            String newLine = ds.readLine();
            while (!newLine.isEmpty()) {
                int newInt = Integer.parseInt(newLine);
                dos.writeInt(newInt);
                newLine = ds.readLine(); }
        } catch (Exception e) {
            System.out.println("Произошла ошибка " +
e.getMessage()); }
        finally {
            dos.close();
            ds.close();
        }
    }
    public static void CopyFile(FileInputStream fis,
FileOutputStream fos) throws IOException
    {
        DataInputStream dis = new DataInputStream(fis);
        DataOutputStream dos = new DataOutputStream(fos);
        while(true)
        { try {
            int newInt = dis.readInt();
            dos.writeInt(newInt);
            System.out.println(String.format("%d\n", newInt));
        } catch (EOFException e) {
            break;
        } catch (Exception e) {
            System.out.println("Произошла ошибка " +
e.getMessage());
        } }
    }
}
} //конец программы

```

## **5. Создать файл на диске, ввести заданное с клавиатуры**

**количество строк текста и записать их в файл в формате UTF-8.**

```
/*Создать файл на диске, ввести заданное с клавиатуры
количество строк текста и записать их в файл в формате UTF-8.*/
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;
public class KlassFile5 {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.print("введите имя файла => "); String
            fname=sc.nextLine();
        try{ // создается файл
            File f1 = GetFileByName(fname);
            System.out.println("Полный путь файла: "+
f1.getAbsolutePath());
            System.out.print("Введите количество строк для записи
в файл => ");
            int n=sc.nextInt();
            // создается поток для записи с учетом типа данных в
DataOutputStream, и ему в качестве параметра передается поток
FileOutputStream
            DataOutputStream dout=
                new DataOutputStream( new
FileOutputStream(f1));
            sc.nextLine(); //очистка буфера
            for (int i = 0; i < n; i++) {
                System.out.print("файл введите строку для записи в
файл => ");
                String s=sc.nextLine();
                dout.writeUTF(s );
                //или dout.writeUTF(s +"\n" ); запись отдельных
строк
            }
            dout.flush(); // дописываем несохраненные данные на
диск
        }
    }
}
```



```

        dout.close(); // закрываем поток, чтение и вывод данных
из созданного файла
        DataInputStream dIn=new DataInputStream(new
FileInputStream(f1));
        while (true) {
            System.out.println(dIn.readUTF());
        }
    }catch (Exception e) {
        System.out.println(""+e);
    } }
    private static String DirPath = "C:\\Test\\%s";
    public static File GetFileByName(String filename) throws
IOException {
        File f1=new File(String.format(DirPath, filename));
        f1.createNewFile();
        return f1;
    }}//конец программы

```

#### ***6. Чтение из одного файла и запись в другой файл данных посимвольно.***

```

/*чтение из одного файла и запись в другой файл данных
посимвольно. */
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Reader;
import java.io.Writer;
public class KlassFile6
{
    public static void main(String[] args) throws IOException {
        FileReader in=null; // можно сразу записать
        FileWriter out =null; // можно сразу записать
        try {
            in = new FileReader(GetFileByName("myFile61.txt")); //
файл для чтения
            out= new FileWriter(GetFileByName("myFile62.txt"),
true); // файл для записи
            // данные считываются и записываются побайтно, как и
для InputStream/OutputStream
            int oneByte; // переменная, в которую считываются

```

данные

```
        while ((oneByte = in.read()) != -1) {
            // out.write((char)oneByte); // запись с
уничтожением ранее существующих данных
            out.append((char)oneByte); // запись с добавлением
данных в конец
            System.out.print((char)oneByte);
        }
    } catch (IOException e) {
        System.out.println("Ошибка!!!! ");
    }
    finally{
        in.close();
        out.close();
    }
}
private static String DirPath = "C:\\\\Test\\\\%s";
public static File GetFileByName(String filename) throws
IOException {
    File f1=new File(String.format(DirPath, filename));
    f1.createNewFile();
    return f1;
}
} //конец программы
```

### ***7. Чтение из одного файла и запись в другой файл данных построчно с использованием буфера в 1 килобайт.***

```
/*чтение из одного файла и запись в другой файл данных построчно с
использованием буфера в 1 килобайт.*/
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Reader;
import java.io.Writer;
public class KlassFile7 { public static void main(String[] args)
throws IOException {
```

```

        BufferedReader in=null;  BufferedWriter out =null;
        try { in = new BufferedReader(new
FileReader(GetFileByName("MyFile611.txt")), 1024);
            out= new BufferedWriter(new
FileWriter(GetFileByName("MyFile612.txt")));
            int lineCount = 0;
            String res;
            while ((res = in.readLine()) != null) { lineCount++;
                System.out.println(lineCount + ": " + res);
                out.write(res);
                out.newLine(); }
        } catch (IOException e) {  System.out.println("Ошибка!!!!
");
        } finally{
            in.close();
            out.flush();
            out.close(); } }
    private static String DirPath = "D:\\Test\\%s";
    public static File GetFileByName(String filename) throws
IOException {
        File f1=new File(String.format(DirPath, filename));
        f1.createNewFile();
        return f1;    }
} //конец программы

```

**8. Прочитать и вывести на экран информацию из трех источников: файла на диске, интернет-страницы и массива данных типа byte. Указать кодировку, поддерживающую кириллицу.**

```

/* Прочитать и вывести на экран информацию из трех источников:
файла на диске, интернет-страницы и массива данных типа byte.
Указать кодировку, поддерживающую кириллицу. */
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.ByteArrayInputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;

```

```

import java.io.InputStreamReader;
import java.io.Reader;
import java.io.Writer;
import java.net.URL;

public class KlassFile8 {
    public static void readAllByByte( Reader in) throws
IOException {
        while (true) {
            int oneByte = in.read(); // читает 1 байт
            if (oneByte != -1) { // признак конца файла
                System.out.print((char) oneByte);
            }
            else {
                System.out.print("\n" + " конец ");
                break;
            }
        }
    }
    public static void main(String[] args) {
        try {
            // с потоком связан файл
            InputStream inFile = new
FileInputStream("D:\\MyFile1.txt"); // байтовый
            // поток
            Reader rFile= new InputStreamReader(inFile,"cp1251");
// символный
            // поток
            readAllByByte(rFile);
            System.out.print("\n\n\n");
            inFile.close();
            rFile.close();
            // с потоком связана интернет-страница
            //передается «русская» кодировка
            InputStream inUrl = new
URL("http://google.com").openStream(); // байтовый поток
            Reader rUrl=new InputStreamReader(inUrl, "cp1251");
//символьный поток
            readAllByByte(rUrl);
            System.out.print("\n\n\n");
            inUrl.close();

```

```

        rUrl.close();
        // С потоком связан массив типа byte
        InputStream inArray = new ByteArrayInputStream( new
byte[] {5, 8, 3, 9, 11});
        Reader rArray=new
InputStreamReader(inArray,"cp1251" ); // символный поток
        readAllByByte(rArray);
        System.out.print("\n\n\n");
        inArray.close();
        rArray.close();
    }
    catch (IOException e) {
        System.out.println("Ошибка: "+ e);
    }
}
//конец программы

```

***9. Чтение из одного файла и запись в другой файл данных построчно с использованием буферизации символьных потоков основанных на байтовых файловых потоках.***

*/\*Чтение из одного файла и запись в другой файл данных построчно с использованием буферизации символьных потоков основанных на байтовых файловых потоках.\*/*

```

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.File;
import java.io.IOException;

public class KlassFile9 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = null;
        BufferedWriter bw = null;
        try { // -оздание потоков длѧ чтения и записи с нужной
кодировкой
            br = new BufferedReader( new InputStreamReader( new

```

```

FileInputStream(GetFileByName("MyFile1.txt")), "cp1251"));
        bw = new BufferedWriter( new OutputStreamWriter( new
FileOutputStream(GetFileByName("MyFile2.txt")), "cp1251"));
        int lineCount = 0; // счетчик строк
        String s;
        while ((s = br.readLine()) != null) {
            lineCount++; System.out.println(lineCount + ": " +
s);
            bw.write(lineCount + ": " + s); // запись без
перевода строки
            bw.newLine(); // принудительный переход на новую
строку
        }
    } catch (IOException e)
{ System.out.println("Ошибка!!!!!!!!!!");
    }
    finally{
        br.close();
        bw.flush();
        bw.close();
    }
}

private static String DirPath = "C:\\Test\\%s";
public static File GetFileByName(String filename) throws
IOException {
    File f1=new File(String.format(DirPath, filename));
    f1.createNewFile();
    return f1;
}
} //конец программы

```

**10. Выполнить чтение из одного файла и запись в другой файл с использованием класса *PrintWriter*.**

```

/*Выполнить чтение из одного файла и запись в другой файл с
использованием класса PrintWriter*/
import java.io.BufferedReader
import java.io.FileInputStream;
import java.io.PrintWriter;
import java.io.File;
import java.io.IOException;

```

```

import java.io.InputStreamReader;
public class KlassFile10 {
    public static void main(String[] args) throws IOException {
        BufferedReader br = null;
        PrintWriter out = null;
        try { // Создание потоков
            br = new BufferedReader(
                new InputStreamReader(
                    new
FileInputStream(GetFileByName("MyFile1.txt")), "cp1251"));
            out = new
PrintWriter(GetFileByName("MyFile2.txt"), "cp1251"); //
Переписывание информации из одного файла в другой
            int lineCount = 0; String s;
            while ((s = br.readLine()) != null) {
                lineCount++; out.println(lineCount + ": " + s);
            }
        }
        catch (IOException e) {
            System.out.println("Ошибка !!!!!!!!!");
        }
        finally{
            br.close();
            out.flush();
            out.close();
        }
    }

    private static String DirPath = "C:\\Test\\%s";
    public static File GetFileByName(String filename) throws
IOException {
        File f1=new File(String.format(DirPath, filename));
        f1.createNewFile();
        return f1;
    }

} //конец программы

```

## ***Выполнение самостоятельного задания.***

- 1. Создать проект, позволяющий из одного, предварительно созданного программными средствами файла, переписать данные, соответствующие условию - в исходном файле содержится две строки в формате UTF-8 и 5 чисел типа double. В результирующий файл переписать вторую строку и положительные числа.***

```
/*Создать проект, позволяющий из одного, предварительно
созданного программными средствами файла, переписать данные,
соответствующие условию - в исходном файле содержится две
строки в формате UTF-8 и 5 чисел типа double. В
результирующий файл переписать вторую строку и положительные
числа.*/
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.File;
import java.io.IOException;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.PrintWriter;
import java.util.Random;
import java.util.UUID;
public class KlassFile11 {
    private static int UTF_COUNT = 3;
    private static int DBL_COUNT = 5;
    public static void main(String[] args) throws IOException
    {
        DataInputStream in = null;
        PrintWriter out = null;
        try {
            in = new DataInputStream(new
FileInputStream(PrepareFile(GetFileName("Task2Input.txt"))))
);
            out = new
PrintWriter(GetFileName("Task2Output.txt"), "cp1251");
            for (int i = 0; i < UTF_COUNT; i++) {
                String string = in.readUTF();
                if (i==1) {
                    out.println(string);
                }
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



```

        }
        System.out.println("Строка №" + i+1 + ": " +
string);
    }
    for (int i = 0; i < DBL_COUNT; i++) {
        double dbl = in.readDouble();
        if (dbl > 0) {
            out.println(dbl);
        }
        System.out.println("Число №" + i+1 + ": " +
dbl);
    }
}
}
catch (Exception e) {
    System.out.println("Error: " + e.getMessage());
}
finally {
    in.close();
    out.flush();
    out.close();
}
}

private static String DirPath = "C:\\Test\\%s";
public static File GetFileByName(String filename) throws
IOException {
    File f1=new File(String.format(DirPath, filename));
    f1.createNewFile();
    return f1;
}
public static File PrepareFile(File file) throws
IOException {
    DataOutputStream in = null;
    try {
        in = new DataOutputStream(new
FileOutputStream(file));
        Random r = new Random();
        for (int i = 0; i < UTF_COUNT; i++) {
            in.writeUTF(UUID.randomUUID().toString());
        }
    }
}

```

```

        for (int i = 0; i < DBL_COUNT; i++) {
            in.writeDouble(-1000 + 2000 *
r.nextDouble());
        }
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
    finally {
        if (in != null) {
            in.close();
        }
    }
    return file;
}
} //конец программы

```

2. *Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (Пушкин, Лермонтов или другой российский классик на Ваш вкус), построчно переписать в другой текстовый файл слова начинающиеся с согласных букв..*

*Требования: – слова из предложения выделять методом split();*

*– в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;*

*– для каждой строки указать количество выбранных слов.*

*/\*Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (Пушкин, Лермонтов или другой российский классик на Ваш вкус), построчно переписать в другой текстовый файл слова, начинающиеся с согласных букв.*

*Требования:*

*– слова из предложения выделять методом split();*

*– в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;*

*– для каждой строки указать количество выбранных слов.\*/\**

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;

```

```

import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.util.ArrayList;
public class KlassFile12 {
    public static void main(String[] args) throws IOException {
        BufferedReader in = null;
        PrintWriter out = null;
        int i = 0;
        try {
            in = new BufferedReader( new InputStreamReader( new
FileInputStream(GetFileByName("Task3_Input.txt")), "cp1251"));
            out = new
PrintWriter(GetFileByName("Task3_Output.txt"), "cp1251");
            String line;
            while ((line = in.readLine()) != null) {
                i++;
                for (String string :
GetWords(line/*.replaceAll(",|.", "")*/)) {
                    System.out.println(i+": "+string);
                    out.println(i+": "+string);
                }
                line = in.readLine();
            }
        }
        catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
        finally {
            in.close();
            out.flush();
            out.close();
        }
    }

    private static final String DirPath = "C:\\\\Test\\\\%s";
    public static File GetFileByName(String filename) throws
IOException
    {
        File f1=new File(String.format(DirPath, filename));
    }
}

```

```

        return f1;
    }

    private static ArrayList<String> GetWords(String inStr)
    {
        ArrayList<String> res = new ArrayList<String>();
        for (String string : inStr.split(" ")) {
            if (StartsWithConsonant(string)) {
                res.add(string);
            }
        }
        return res;
    }

    private static final String ConsonantList =
"бвгджзйклмнпрстфхцчщц";
    private static boolean StartsWithConsonant(String inStr)
    {
        return ConsonantList.contains(inStr.subSequence(0, 1));
    }
}

```

### ***Вывод:***

В ходе работы были выполнены 10 решённых задач и разработано два небольших проекта. Листинги программ представлены в теле лабораторной работы №2. Программы скомпилированы и отлажены, файлы классов будут приложены в архив с работой.

При помощи инструментов, создаваемых в ходе работы, можно свободно работать с файлами и папками, заполняя данными разного типа.

Используемые при работе с файлами и папками, методы можно применять при разработке модулей программного обеспечения, а также для автоматизации обработки данных.