“大数据工程”课程实验报告

题目：HDFS 的编程实验	学号姓名：21377061 范春	日期：2024.03.12

实验环境：

1、虚拟机软件：VMvare

2、Hdoop 版本：3.1.3

3、Java 版本：Oracle JDK 1.8

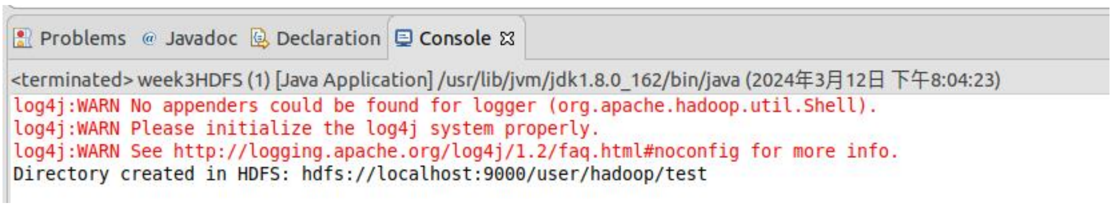
4、Java IDE：Eclipse

实验内容与完成情况：

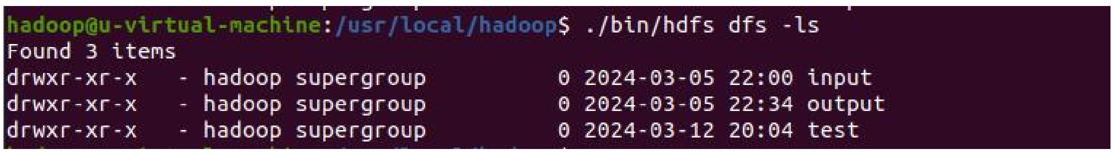
一、Eclipse 直接编译：

1、创建新的文件目录

```
    public static void mkDir(String dirName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path dirPath = new Path(dirName);

            if (fs.exists(dirPath)) {
                System.out.println("Directory already exists in HDFS.");
            } else {
                fs.mkdirs(dirPath);
                System.out.println("Directory created in HDFS: " + dirName);
            }

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```



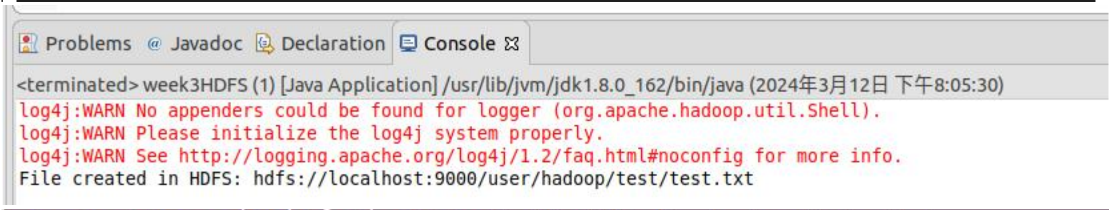
```
<terminated> week3HDFS (1) [Java Application] /usr/lib/jvm/jdk1.8.0_162/bin/java (2024年3月12日 下午8:04:23)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
Directory created in HDFS: hdfs://localhost:9000/user/hadoop/test
```



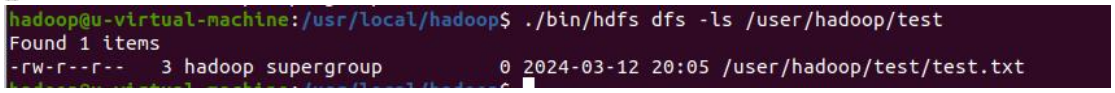
```
hadoop@u-virtual-machine:/usr/local/hadoop$ ./bin/hdfs dfs -ls
Found 3 items
drwxr-xr-x   - hadoop supergroup          0 2024-03-05 22:00 input
drwxr-xr-x   - hadoop supergroup          0 2024-03-05 22:34 output
drwxr-xr-x   - hadoop supergroup          0 2024-03-12 20:04 test
```

2、创建新的文件

```
    public static void createFile(String fileName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path filePath = new Path(fileName);

            if (!fs.exists(filePath.getParent())) {
                mkDir(filePath.getParent().toString());
            }

            fs.create(filePath).close();
            System.out.println("File created in HDFS: " + fileName);

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```



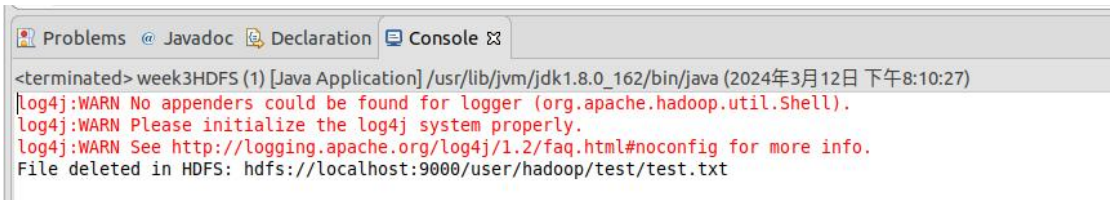
```
<terminated> week3HDFS (1) [Java Application] /usr/lib/jvm/jdk1.8.0_162/bin/java (2024年3月12日 下午8:05:30)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/test.txt
```



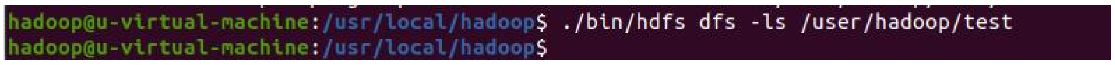
```
hadoop@u-virtual-machine:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop/test
Found 1 items
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:05 /user/hadoop/test/test.txt
```

3、删除文件

```
    public static void delFile(String fileName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path filePath = new Path(fileName);

            if (!fs.exists(filePath)) {
                throw new Exception("File does not exist in HDFS: " + fileName);
            }

            fs.delete(filePath, true);
            System.out.println("File deleted in HDFS: " + fileName);

            fs.close();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
```



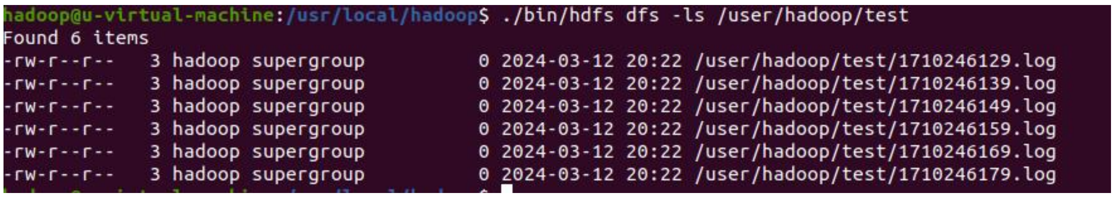
```
<terminated> week3HDFS (1) [Java Application] /usr/lib/jvm/jdk1.8.0_162/bin/java (2024年3月12日 下午8:10:27)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
File deleted in HDFS: hdfs://localhost:9000/user/hadoop/test/test.txt
```



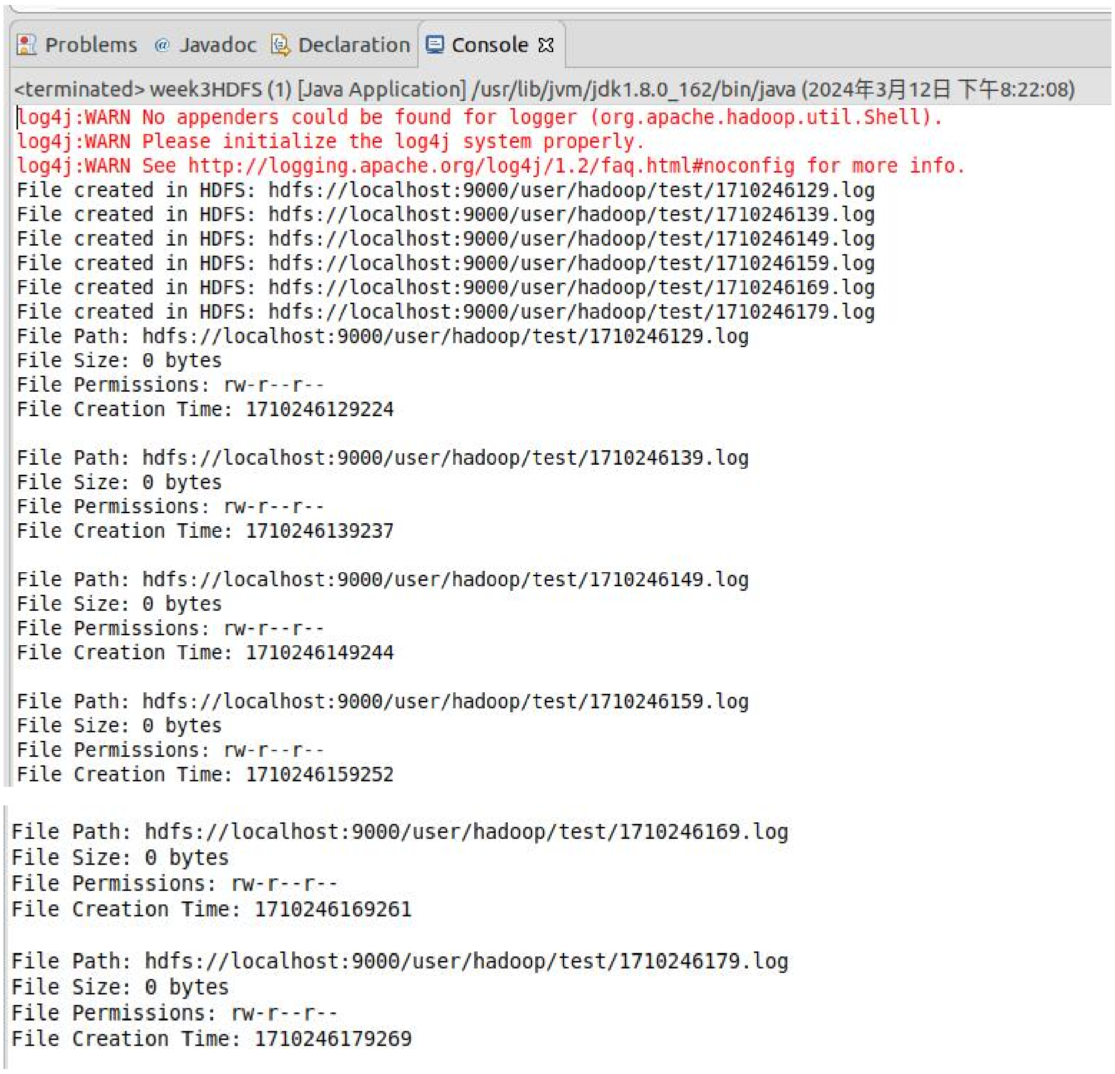
```
hadoop@u-virtual-machine:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop/test
hadoop@u-virtual-machine:/usr/local/hadoop$
```

4、自动批量创建文件并获取文件相关元信息

```
    public static void batchCreateReadFiles(String dirName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path dirPath = new Path(dirName);

            long startTime = Instant.now().getEpochSecond();
            long duration = 60; // 1 minute

            while (Instant.now().getEpochSecond() - startTime < duration) {
                String fileName = dirName + "/" + Instant.now().getEpochSecond() +
".log";
                fs.create(new Path(fileName)).close();
                System.out.println("File created in HDFS: " + fileName);

                Thread.sleep(10000); // 10 seconds
            }

            FileStatus[] fileStatuses = fs.listStatus(dirPath);
            for (FileStatus status : fileStatuses) {
                System.out.println("File Path: " + status.getPath());
                System.out.println("File Size: " + status.getLen() + " bytes");
                System.out.println("File Permissions: " + status.getPermission());
                System.out.println("File Creation Time: " + status.getModificationTime());
                System.out.println();
            }

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
            }
        }
```

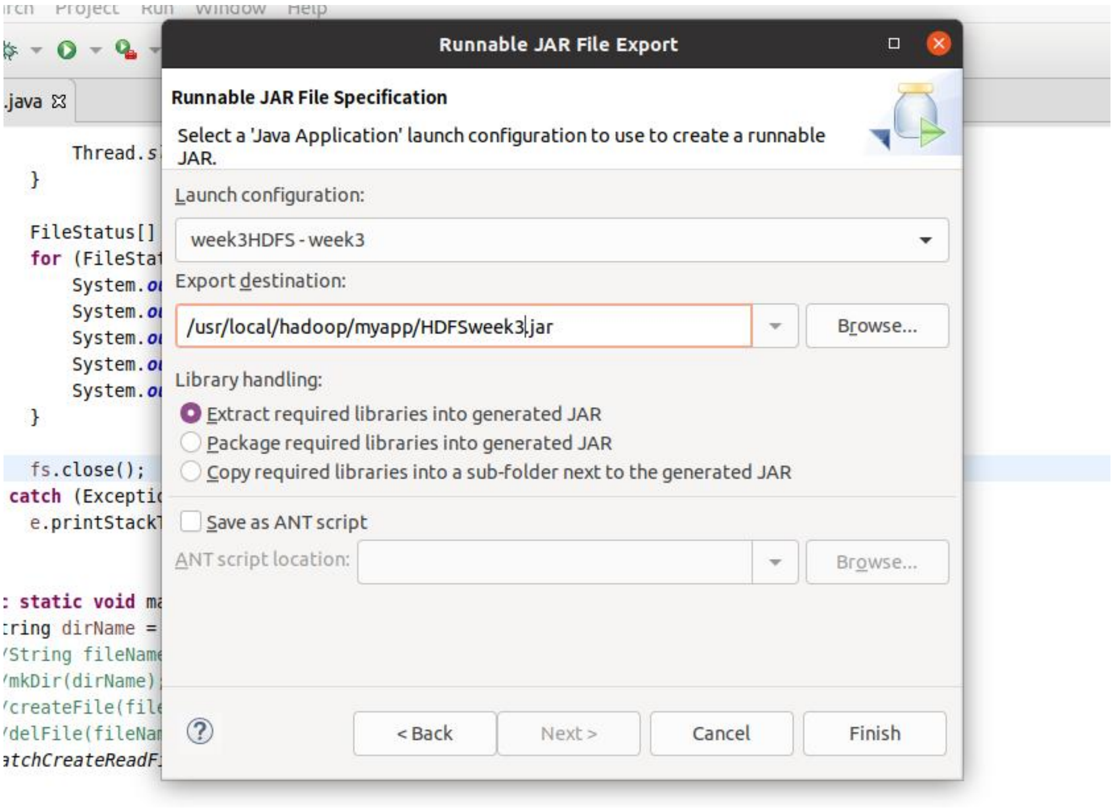


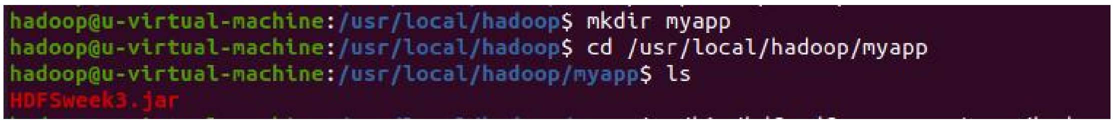
```
hadoop@u-virtual-machine:/usr/local/hadoop$ ./bin/hdfs dfs -ls /user/hadoop/test
Found 6 items
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246129.log
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246139.log
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246149.log
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246159.log
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246169.log
-rw-r--r--   3 hadoop supergroup          0 2024-03-12 20:22 /user/hadoop/test/1710246179.log
```



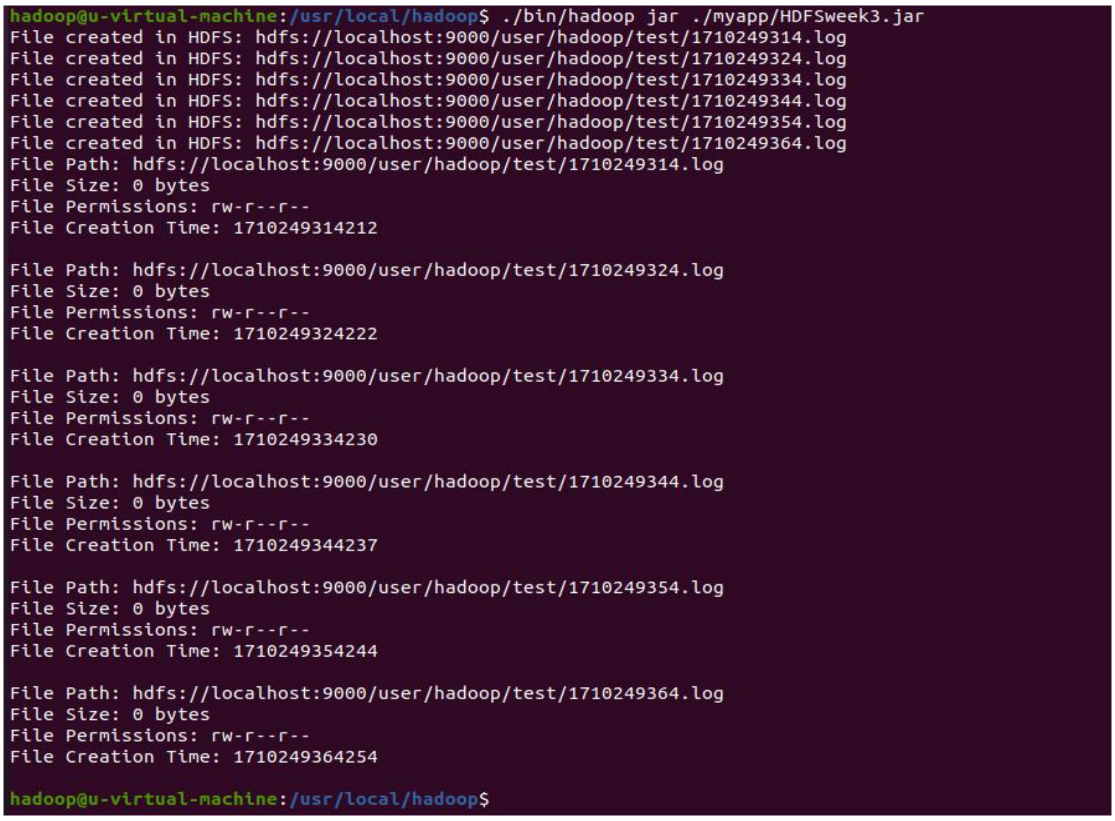
```
<terminated> week3HDFS (1) [Java Application] /usr/lib/jvm/jdk1.8.0_162/bin/java (2024年3月12日 下午8:22:08)
log4j:WARN No appenders could be found for logger (org.apache.hadoop.util.Shell).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246129.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246139.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246149.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246159.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246169.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710246179.log
File Path: hdfs://localhost:9000/user/hadoop/test/1710246129.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246129224

File Path: hdfs://localhost:9000/user/hadoop/test/1710246139.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246139237

File Path: hdfs://localhost:9000/user/hadoop/test/1710246149.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246149244

File Path: hdfs://localhost:9000/user/hadoop/test/1710246159.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246159252

File Path: hdfs://localhost:9000/user/hadoop/test/1710246169.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246169261

File Path: hdfs://localhost:9000/user/hadoop/test/1710246179.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710246179269
```

二、将 Java 应用程序生成 JAR 包，部署到 Hadoop 平台上进行运行


Runnable JAR File Export
Runnable JAR File Specification
Select a 'Java Application' launch configuration to use to create a runnable JAR.
Launch configuration:
week3HDFS - week3
Export destination:
/usr/local/hadoop/myapp/HDFSweek3.jar
Browse...
Library handling:
Extract required libraries into generated JAR
Package required libraries into generated JAR
Copy required libraries into a sub-folder next to the generated JAR
Save as ANT script
ANT script location:
Browse...
< Back
Next >
Cancel
Finish



```
hadoop@u-virtual-machine:/usr/local/hadoop$ mkdir myapp
hadoop@u-virtual-machine:/usr/local/hadoop$ cd /usr/local/hadoop/myapp
hadoop@u-virtual-machine:/usr/local/hadoop/myapp$ ls
HDFSweek3.jar
```



```
hadoop@u-virtual-machine:/usr/local/hadoop$ ./bin/hadoop jar ./myapp/HDFSweek3.jar
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249314.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249324.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249334.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249344.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249354.log
File created in HDFS: hdfs://localhost:9000/user/hadoop/test/1710249364.log
File Path: hdfs://localhost:9000/user/hadoop/test/1710249314.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249314212

File Path: hdfs://localhost:9000/user/hadoop/test/1710249324.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249324222

File Path: hdfs://localhost:9000/user/hadoop/test/1710249334.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249334230

File Path: hdfs://localhost:9000/user/hadoop/test/1710249344.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249344237

File Path: hdfs://localhost:9000/user/hadoop/test/1710249354.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249354244

File Path: hdfs://localhost:9000/user/hadoop/test/1710249364.log
File Size: 0 bytes
File Permissions: rw-r--r--
File Creation Time: 1710249364254

hadoop@u-virtual-machine:/usr/local/hadoop$
```

出现的问题：导出 jar 包时，不知道为什么 launch configuration 选项下有两个选项（week3HDFS-week3 和 week3HDFS-week3（1）），我一开始选择了 week3HDFS-week3，导出的 jar 包在运行时出现了报错：RunJar jarFile [mainClass] args...以为是技术问题，寻找了很多方法依然没有解决，后来重新选择 week3HDFS-week3（1）导出 jar 包就能正常运行了。

完整源码如下所示：

```
package week3;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.*;
import java.time.Instant;

public class week3HDFS {
    public static void mkDir(String dirName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path dirPath = new Path(dirName);

            if (fs.exists(dirPath)) {
                System.out.println("Directory already exists in HDFS.");
            } else {
                fs.mkdirs(dirPath);
                System.out.println("Directory created in HDFS: " + dirName);
            }

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void createFile(String fileName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path filePath = new Path(fileName);

            if (!fs.exists(filePath.getParent())) {
                mkDir(filePath.getParent().toString());
            }

            fs.create(filePath).close();
            System.out.println("File created in HDFS: " + fileName);

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void delFile(String fileName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path filePath = new Path(fileName);

            if (!fs.exists(filePath)) {
                throw new Exception("File does not exist in HDFS: " + fileName);
            }

            fs.delete(filePath, true);
            System.out.println("File deleted in HDFS: " + fileName);

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void batchCreateReadFiles(String dirName) {
        try {
            Configuration conf = new Configuration();
            conf.set("fs.defaultFS","hdfs://localhost:9000");
            conf.set("fs.hdfs.impl","org.apache.hadoop.hdfs.DistributedFileSystem");
            FileSystem fs = FileSystem.get(conf);
            Path dirPath = new Path(dirName);

            long startTime = Instant.now().getEpochSecond();
            long duration = 60; // 1 minute

            while (Instant.now().getEpochSecond() - startTime < duration) {
```

```java
                String fileName = dirName + "/" + Instant.now().getEpochSecond() + ".log";
                fs.create(new Path(fileName)).close();
                System.out.println("File created in HDFS: " + fileName);

                Thread.sleep(10000); // 10 seconds
            }

            FileStatus[] fileStatuses = fs.listStatus(dirPath);
            for (FileStatus status : fileStatuses) {
                System.out.println("File Path: " + status.getPath());
                System.out.println("File Size: " + status.getLen() + " bytes");
                System.out.println("File Permissions: " + status.getPermission());
                System.out.println("File Creation Time: " + status.getModificationTime());
                System.out.println();
            }

            fs.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        String dirName = "hdfs://localhost:9000/user/hadoop/test";
        //String fileName = "hdfs://localhost:9000/user/hadoop/test/test.txt";
        //mkDir(dirName);
        //createFile(fileName);
        //delFile(fileName);
        batchCreateReadFiles(dirName);
    }
}
```