

Java Text File 읽기

데렉토리 만드는 방법입니다.

```
import java.io.File;
import java.util.Arrays;

public class MakeDrectoryEx {
    public static void main(String[] args) {
        makeDrectory("data1");
        makeDrectory("data2/f1/f2");
        makeDrectory("data3\\f1\\f2");
        makeDrectory2("data4\\f1\\f2");
        makeDrectory2("data5/f1/f2");
    }
    private static void makeDrectory(String path) {
        String[] paths = null;
        if(path.contains("/")) {
            paths = path.split("/");
        }else if(path.contains("\\")) {
            paths = path.split("\\\\");
        }else {
            paths = new String[1];
            paths[0] = path;
        }
        System.out.println(Arrays.toString(paths));
        String tempPath="";
        for(int i=0;i<paths.length;i++) {
            tempPath += paths[i];
            File file = new File(tempPath);
            if(!file.exists()) {
                file.mkdir();
            }
            tempPath += File.separator;
        }
    }
    private static void makeDrectory2(String path) {
        File file = new File(path);
        if(!file.exists()) {
            file.mkdirs();
        }
    }
}
```

실행해보고 프로젝트에서 새로고침을 하면 현재 프로젝트 밑에 폴더들이 생긴것을 확인할 수 있습니다.

실습에 사용될 텍스트 파일 입니다.

bugs.txt

```
Assasin bug, Avondale spider, Backswimmer,
Bamboo moth, Banana moth, Bed bug,
```

Black cocroach, Blue moon, Bumble Bee,
Carpenter Bee, Cattle tick, Cave Weta,
Cicada, Cinnibar, Click beetle, Clothes moth,
Codling moth, Centipede, Earwig, Eucalypt longhorn beetle,
Field Grasshopper, Garden slug, Garden soldier,
German cockroach, German wasp, Giant dragonfly,
Giraffe weevil, Grass grub, Grass looper,
Green planthopper, Green house spider, Gum emperor,
Gum leaf skeletoniser, Hornet, Mealybug,
Mites, Mole Cricket, Monarch butterfly,
Mosquito, Silverfish, Wasp,
Water boatman, Winged weta, Wolf spider,
Yellow Jacket, Yellow Admiral

NationalAnthem.txt

동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세
무궁화 삼천리 화려강산 대한사람 대한으로 길이 보전하세
남산위에 저 소나무 철갑을 두른듯 바람서리 불변함은 우리기상 일세
무궁화 삼천리 화려강산 대한사람 대한으로 길이보전하세
가을하늘 공활한데 높고 구름없이 밝은달은 우리가슴 일편단심일세
무궁화 삼천리 화려강산 대한사람 대한으로 길이보전하세
이 기상과 이 맘으로 충성을 다하여 괴로우나 즐거우나 나라사랑하세
무궁화 삼천리 화려강산 대한사람 대한으로 길이보전하세

절대 경로를 알아내는 코드 입니다.

```
import java.io.File;

public class AbsolutePath {
    public static void main(String[] args) {
        File curDir = new File("");
        System.out.println("현재 프로젝트 절대 경로 : " +
curDir.getAbsolutePath());
        curDir = new File("src/main/resources/NationalAnthem.txt");
        System.out.println("현재 프로젝트 안의 파일 절대 경로 : " +
curDir.getAbsolutePath());
    }
}
```

결과

현재 프로젝트 절대 경로 : D:\SpringJava\workspace\RegularExpression
현재 프로젝트 안의 파일 절대 경로 :
D:\SpringJava\workspace\RegularExpression\src\main\resources\NationalAnthem.txt

1. FileInputStream을 이용한 텍스트 파일 읽기

```
import java.io.FileInputStream;
import java.io.IOException;
// FileInputStream을 이용한 텍스트 파일 읽기
public class Ex01_TextFileReader1 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try(FileInputStream fis = new FileInputStream(fileName)){
            StringBuilder sb = new StringBuilder();
            int ch=0;
            while((ch=fis.read())>0) {
                sb.append((char)ch);
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println();

        byte[] bytes = new byte[1024];
        try(FileInputStream fis = new FileInputStream(fileName)){
            StringBuilder sb = new StringBuilder();
            int len=0;
            while((len=fis.read(bytes))>0) {
                sb.append(new String(bytes, 0, len));
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

2. FileReader를 이용한 텍스트 파일 읽기

```
import java.io.FileReader;
import java.io.IOException;
// FileReader를 이용한 텍스트 파일 읽기
public class Ex02_TextFileReader2 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try(FileReader fr = new FileReader(fileName)){
            StringBuilder sb = new StringBuilder();
            int ch=0;
            while((ch=fr.read())>0) {
                sb.append((char)ch);
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println();

        char[] chars = new char[1024];
        try(FileReader fr = new FileReader(fileName)){
```

```

        StringBuilder sb = new StringBuilder();
        int len=0;
        while((len=fr.read(chars))>0) {
            sb.append(new String(chars, 0, len));
        }
        System.out.println(sb);
    }catch (IOException e) {
        e.printStackTrace();
    }
}
}

```

3. BufferedReader를 이용한 텍스트 파일 읽기

```

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
// BufferedReader를 이용한 텍스트 파일 읽기
public class Ex03_TextFileReader3 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try(BufferedReader br = new BufferedReader(new FileReader(fileName))){
            StringBuilder sb = new StringBuilder();
            String line="";
            while((line=br.readLine())!=null) {
                sb.append(line);
                sb.append(System.lineSeparator());
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

4. InputStreamReader를 이용한 텍스트 파일 읽기

```

import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
// InputStreamReader를 이용한 텍스트 파일 읽기
public class Ex04_TextFileReader4 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try(InputStreamReader isr = new InputStreamReader(new
FileInputStream(fileName))){
            StringBuilder sb = new StringBuilder();
            int n=0;
            while((n=isr.read())>0) {
                sb.append((char)n);
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
        System.out.println();
    }
}

```

```

        char[] chars = new char[1024];
        try(InputStreamReader isr = new InputStreamReader(new
FileInputStream(fileName))){
            StringBuilder sb = new StringBuilder();
            int len=0;
            while((len=isr.read(chars))>0) {
                sb.append(new String(chars, 0, len));
            }
            System.out.println(sb);
        }catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

5. Scanner를 이용한 텍스트 파일 읽기

```

import java.io.File;
import java.util.Scanner;
// Scanner를 이용한 텍스트 파일 읽기
public class Ex05_TextFileReader5 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try(Scanner sc = new Scanner(new File(fileName))){
            StringBuilder sb = new StringBuilder();
            while(sc.hasNextLine()) {
                sb.append(sc.nextLine());
                sb.append(System.lineSeparator());
            }
            System.out.println(sb);
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

6. Files.newBufferedReader를 이용한 텍스트 파일 읽기

```

import java.io.BufferedReader;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;

// Files.newBufferedReader를 이용한 텍스트 파일 읽기
public class Ex06_TextFileReader6 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";

        try (BufferedReader br = Files.newBufferedReader(Paths.get(fileName),
StandardCharsets.UTF_8)) {
            String line;
            while ((line = br.readLine()) != null) {
                System.out.println(line);
            }
        }
    }
}

```

```

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

7. Files.lines을 이용한 텍스트 파일 읽기

```

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
// Files.lines을 이용한 텍스트 파일 읽기
public class Ex07_TextFileReader7 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";
        try {

Files.lines(Paths.get(fileName)).forEachOrdered(System.out::println);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

8. Files.readAllBytes을 이용한 텍스트 파일 읽기

```

import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
// Files.readAllBytes을 이용한 텍스트 파일 읽기
public class Ex08_TextFileReader8 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";

        try {
            byte[] data = Files.readAllBytes(Paths.get(fileName));
            String content = new String(data);
            System.out.println(content);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

9. Files.readString을 이용한 텍스트 파일 읽기

```

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
// Files.readString을 이용한 텍스트 파일 읽기
public class Ex09_TextFileReader9 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";

```

```

        try {
            String content = Files.readString(Paths.get(fileName),
StandardCharsets.UTF_8);
            System.out.println(content);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

10. Files.readAllLines을 이용한 텍스트 파일 읽기

```

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.List;
// Files.readAllLines을 이용한 텍스트 파일 읽기
public class Ex10_TextFileReader10 {
    public static void main(String[] args) {
        String fileName = "src/main/resources/NationalAnthem.txt";

        try {
            List<String> lines = Files.readAllLines(Paths.get(fileName),
StandardCharsets.UTF_8);
            for(String line : lines) System.out.println(line);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Java File 복사하기

1. FileInputStream/FileOutputStream을 이용한 복사(JDK 1.7 전 방식)

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Ex01_CopyingFile1 {
    public static void main(String[] args) {
        String source = "src/main/resources/bugs.txt";
        String dest = "data/resources/bugs2.txt";
        makeDirectory("data/resources/");
        FileInputStream fis = null;
        FileOutputStream fos = null;
        try {
            fis = new FileInputStream(source);
            fos = new FileOutputStream(dest);
            byte[] buffer = new byte[1024];
            int length;

```

```

        while ((length = fis.read(buffer)) > 0) {
            fos.write(buffer, 0, length);
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            fis.close();
            fos.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
File sourceFile = new File(source);
File destFile = new File(dest);
System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");
}
private static void makeDrectory(String path) {
    File file = new File(path);
    if(!file.exists()) {
        file.mkdirs();
    }
}
}
}

```

2. FileInputStream/FileOutputStream을 이용한 복사(JDK 1.7 이후 방식)

```

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class Ex02_CopyingFile2 {
    public static void main(String[] args) {
        String source = "src/main/resources/bugs.txt";
        String dest = "data/resources/bugs2.txt";
        makeDrectory("data/resources/");

        try (FileInputStream fis = new FileInputStream(source);
            FileOutputStream fos = new FileOutputStream(dest)) { // 자동닫기
            byte[] buffer = new byte[1024];
            int length;
            while ((length = fis.read(buffer)) > 0) {
                fos.write(buffer, 0, length);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        File sourceFile = new File(source);
        File destFile = new File(dest);
        System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
        System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");
    }
}

```



```

    }
    private static void makeDirectory(String path) {
        File file = new File(path);
        if(!file.exists()) {
            file.mkdirs();
        }
    }
}

```

3. FileChannel을 이용한 방법

```

import java.io.File;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.nio.channels.FileChannel;

public class Ex03_CopyingFile3 {
    public static void main(String[] args) {
        makeDirectory("data/resources");
        try(RandomAccessFile source = new
RandomAccessFile("src/main/resources/bugs.txt", "r");
        RandomAccessFile dest = new
RandomAccessFile("data/resources/bugs4.txt", "rw");
        FileChannel sfc = source.getChannel();
        FileChannel dfc = dest.getChannel();){
            dfc.transferFrom(sfc, 0, sfc.size());
        } catch (IOException e) {
            e.printStackTrace();
        }
        File sourceFile = new File("src/main/resources/bugs.txt");
        File destFile = new File("data/resources/bugs4.txt");
        System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
        System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");
    }

    private static void makeDirectory(String path) {
        File file = new File(path);
        if (!file.exists()) {
            file.mkdirs();
        }
    }
}

```

4. Files.newInputStream/Files.newOutputStream을 이용한 복사

```

import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;

```

```

public class Ex04_CopyingFile4 {
    public static void main(String[] args) {
        Path source = Paths.get("src/main/resources/bugs.txt");
        Path dest = Paths.get("data/resources/bugs3.txt");
        makeDrectory("data/resources/");

        try (InputStream fis = Files.newInputStream(source);
            OutputStream fos = Files.newOutputStream(dest)) {
            byte[] buffer = new byte[1024];
            int length;
            while ((length = fis.read(buffer)) > 0) {
                fos.write(buffer, 0, length);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        File sourceFile = source.toFile();
        File destFile = dest.toFile();
        System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
        System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");

    }
    private static void makeDrectory(String path) {
        File file = new File(path);
        if(!file.exists()) {
            file.mkdirs();
        }
    }
}

```

5. Java 7 Files.copy()를 이용한 복사

Java 7 Files.copy()에는 파일을 쉽게 복사할 수 있는 방법이 도입되었습니다.

REPLACE_EXISTING 옵션을 지정 하지 않으면 대상 파일이 있으면 복사가 실패 합니다.

Files.copy() 선택적 세 번째 복사 옵션 인수를 취합니다.

options 매개변수에는 다음 중 하나가 포함될 수 있습니다.

REPLACE_EXISTING - 대상 파일이 존재하는 경우 대상 파일이 비어 있지 않은 디렉토리가 아니면 대상 파일이 대체됩니다.

COPY_ATTRIBUTES - 이 파일과 관련된 파일 속성을 대상 파일에 복사하려고 시도합니다.

ATOMIC_MOVE - 파일을 이동합니다.

NOFOLLOW_LINKS - 심볼릭 링크를 따르지 않습니다.

```

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.StandardCopyOption;

public class Ex05_CopyingFile5 {
    public static void main(String[] args) {
        File source = new File("src/main/resources/bugs.txt");
        makeDrectory("data/resources/");
        File dest = new File("data/resources/bugs5.txt");

        try {

```

```

        Files.copy(source.toPath(),
dest.toPath(),StandardCopyOption.REPLACE_EXISTING);
    } catch (IOException e) {
        e.printStackTrace();
    }
    File sourceFile = new File("src/main/resources/bugs.txt");
    File destFile = new File("data/resources/bugs5.txt");
    System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
    System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");

}
private static void makeDrectory(String path) {
    File file = new File(path);
    if(!file.exists()) {
        file.mkdirs();
    }
}
}
}

```

6. Apache Commons IO의 FileUtils.copyFile()를 이용한 복사

pom.xml에 추가

```

commons-io
commons-io
2.5

```

```

import java.io.File;
import java.io.IOException;
import org.apache.commons.io.FileUtils;

public class Ex06_CopyingFile6 {
    public static void main(String[] args) {
        File source = new File("src/main/resources/bugs.txt");
        makeDrectory("data/resources/");
        File dest = new File("data/resources/bugs6.txt");

        try {
            FileUtils.copyFile(source, dest);
        } catch (IOException e) {
            e.printStackTrace();
        }
        File sourceFile = new File("src/main/resources/bugs.txt");
        File destFile = new File("data/resources/bugs6.txt");
        System.out.print(sourceFile.getName() + "(" + sourceFile.length() +
"Byte)를 ");
        System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)
로 복사완료!!");

    }
    private static void makeDrectory(String path) {
        File file = new File(path);
        if(!file.exists()) {
            file.mkdirs();
        }
    }
}

```

```
}  
}
```

7. Google Guava의 Files.copy()를 이용한 복사

pom.xml에 추가

```
com.google.guava  
guava  
28.0-jre
```

```
import java.io.File;  
import java.io.IOException;  
  
import com.google.common.io.Files;  
/*  
Java copying file with Guava  
pom.xml에 추가  
<dependency>  
    <groupId>com.google.guava</groupId>  
    <artifactId>guava</artifactId>  
    <version>28.0-jre</version>  
</dependency>  
*/  
public class Ex07_CopyingFile7 {  
    public static void main(String[] args) {  
        File source = new File("src/main/resources/bugs.txt");  
        makeDirectory("data/resources/");  
        File dest = new File("data/resources/bugs7.txt");  
  
        try {  
            Files.copy(source, dest);  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
  
        File sourceFile = new File("src/main/resources/bugs.txt");  
        File destFile = new File("data/resources/bugs7.txt");  
        System.out.print(sourceFile.getName() + "(" + sourceFile.length() +  
"Byte)를 ");  
        System.out.println(destFile.getName() + "(" + destFile.length() + "Byte)  
로 복사완료!!");  
  
    }  
    private static void makeDirectory(String path) {  
        File file = new File(path);  
        if(!file.exists()) {  
            file.mkdirs();  
        }  
    }  
}
```

