

জাভায় ফাইল নিয়ে কাজ করার

মো: আবদুল কাদের

প্রোগ্রাম নিয়ে কাজ করতে গিয়ে প্রায়ই ফাইলে ইনপুট-আউটপুট সংক্রান্ত কাজগুলো করতে হয়। প্রোগ্রামিংয়ের প্রয়োজনে কখনও ফাইলে তথ্য সংরক্ষণ করতে হয়, আবার কখনও বা ফাইল থেকে তথ্য নিয়ে কাজ করতে বা দেখার প্রয়োজন হতে পারে। এসব কাজ করার জন্য এ লেখায় জাভা প্রোগ্রামিংয়ের মাধ্যমে কীভাবে ফাইলে তথ্য সংরক্ষণ করা যায় এবং সেই ফাইল থেকে কীভাবে তথ্যগুলো দেখা যায়, সে সংক্রান্ত প্রোগ্রাম নিয়ে আলোচনা করা হয়েছে। তবে এখানে বলে রাখা ভালো, আমরা কোনো ডাটাবেজে তথ্য রাখব না। ডাটাবেজে তথ্য রাখার পদ্ধতি ভিন্ন। ডাটাবেজে তথ্য রাখার জন্য ডাটাবেজ দরকার হবে এবং ডাটাবেজে তথ্য সংযোজন বা ইনসার্ট করতে হলে ডাটাবেজ ল্যাঙ্গুয়েজ সম্পর্কে ধারণা থাকতে হবে। এর জন্য ব্যাপক ধারণা থাকা প্রয়োজন। তাই এ সংক্রান্ত প্রোগ্রাম নিয়ে আমরা পরবর্তী কোনো পর্বে আলোচনা করব।

এ লেখার প্রোগ্রামগুলো রান করার পদ্ধতি অন্যান্য জাভা প্রোগ্রামের মতোই। আমরা রান করার জন্য জাভার Jdk1.4 ভার্সন ব্যবহার করব এবং প্রোগ্রামগুলো D:\ ড্রাইভের test ফোল্ডারে সেভ করব।

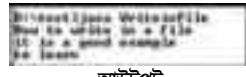
প্রোগ্রামের মাধ্যমে ফাইল তৈরি করে তাতে তথ্য রাখার জন্য নিচের প্রোগ্রামটিকে WriteinFile.java নামে সেভ করতে হবে। প্রোগ্রামটি রান করলে আমাদের ফোল্ডারটিতে abc.txt নামে একটি ফাইল তৈরি হবে, যেখানে একটি বাক্যকে আমরা সংরক্ষণ করব।

```
import java.io.*;
class WriteinFile {
public static void main(String args[]) throws IOException {
String write1 = "How to write in a file\r\n"
+ "it is a good example\r\n"
+ "to learn\r\n\r\n";
FileOutputStream f2 = new FileOutputStream("D:/test/abc.txt");
byte a[] = write1.getBytes();
f2.write(a);
System.out.println(new String(a,0,a.length));
}
}
```



রান করার পদ্ধতি

তারপর ওই বাক্যটিকে স্ক্রিনে দেখাও যাবে। যেমন—

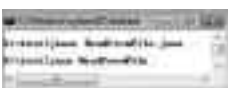


আউটপুট

ফাইলে আগে সংরক্ষিত কোনো তথ্য শুধু পড়ার জন্য নিচের প্রোগ্রামটি

ReadfromFile.java নামে সেভ করতে হবে।

```
import java.io.*;
class ReadfromFile
{
public static void main(String sd[]) throws
IOException,FileNotFoundException
{
String s;
String str="abc.txt";
File f=new File(str);
FileReader is=new FileReader(f);
BufferedReader b=new BufferedReader(is);
while((s=b.readLine())!=null)
{
System.out.println(s);
}
}
}
```



রান করার পদ্ধতি

এই প্রোগ্রামের মাধ্যমে ফাইল থেকে তথ্য নেয়ার জন্য

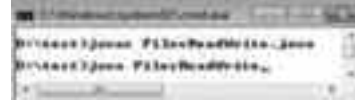
BufferedReader

আউটপুট

নেয়া হয়, যার মাধ্যমে ফাইল থেকে তথ্যগুলো নিয়ে পরবর্তী সময় স্ক্রিনে দেখাবে।

একই সাথে কোনো ফাইলে লেখা এবং তা থেকে পড়ার জন্য নিচের প্রোগ্রামটি FilesReadWrite.java নামে সেভ করতে হবে। FileInputStream-এর মাধ্যমে ফাইলে তথ্য প্রেরণ করা হয় এবং FileOutputStream-এর মাধ্যমে ফাইল থেকে তথ্য পড়া যায়। প্রোগ্রামের শেষ দুটি লাইন File f1=new File("abc.txt"); f1.delete(); এর মাধ্যমে abc.txt ফাইলটিকে মুছে ফেলা হয়।

```
import java.io.FileOutputStream;
import java.io.FileInputStream;
import java.io.File;
import java.io.IOException;
public class FilesReadWrite{
public static void main(String args[])throws IOException{
//creating an output file abc.txt
FileOutputStream os = new FileOutputStream("abc.txt");
String s = "Welcome to File Input Output Stream";
for(int i = 0;i<s.length();i++)
os.write(s.charAt(i));
os.close();
//opening abc.txt for input
FileInputStream is = new FileInputStream("abc.txt");
int ibyts = is.available();
System.out.println("Input Stream has "+ibyts+" available bytes");
byte ibuf[]=new byte[ibyts];
int byrd = is.read(ibuf,0,ibyts);
System.out.println("Number of Bytes read are:"+byrd);
System.out.println("They are:"+new String(ibuf));
is.close();
File f1=new File("abc.txt");
f1.delete();
}
}
```



রান করার পদ্ধতি

ফাইলে র্যানডম অ্যাক্সেস করার জন্য অর্থাৎ সবসময় ডাটা রিড ও রাইট করার জন্য RandomAccessFile ক্লাসটি ব্যবহার করা হয়। এখানে ফাইল

থেকে ডাটা নেয়া বা ফাইলে লেখার তিনটি প্যারামিটার অপশন থাকে। যদি ফাইলে শুধু রিড করার প্যারামিটার দেয়া হয়, তাহলে r ব্যবহার করতে হয়। যেমন— RandomAccessFile ("abc.txt","r"); যদি ফাইলে লেখার প্যারামিটার দেয়া হয় তাহলে w এবং রিড ও রাইট দুটিই প্যারামিটার দেয়া থাকলে rw লেখা হয়।

```
RandomFile.java
import java.lang.System;
import java.io.RandomAccessFile;
import java.io.IOException;
public class RandomFile{
public static void main (String arg[]) throws IOException{
RandomAccessFile rf;
rf=new RandomAccessFile("abc.txt","rw");
rf.writeBoolean(true);
rf.writeInt(67868);
rf.writeChars("J");
rf.writeDouble(678.68);
rf.seek(1);
System.out.println(rf.readInt());
System.out.println(rf.readChar());
System.out.println(rf.readDouble());
rf.seek(0);
System.out.println(rf.readBoolean());
rf.close();
}
}
```

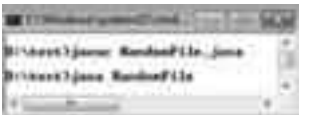
এ লেখায় আলোচিত

প্রোগ্রামগুলোর জন্য একটি Exception ক্লাস ব্যবহার করা হয়েছে, যার নাম IOException। ফাইল নিয়ে কাজ করার সময় অর্থাৎ প্রোগ্রামটি রান করার সময় যদি ফাইল না পাওয়া যায়,

সেটি যে কারণেই হোক যেমন— ফাইল তৈরি না করা বা ফাইলের সঠিক নাম ব্যবহার না করা বা ফাইলে পাঠ ঠিকমতো সেট না করা হয়, তাহলে IOException ঘটতে পারে। সে ক্ষেত্রে প্রোগ্রামটি কী করবে তা নির্ধারণ করার জন্য এটি ব্যবহার করা হয়



আউটপুট



রান করার পদ্ধতি

ফিডব্যাক : balaith@gmail.com