

28/07/2022

Java Notes Day6

Day 6

1. Exception handling(contd.) - Runtime(unchecked) Exceptions
2. File handling
3. Serialization

Exception - Synonymous with runtime error

Keywords - throws, finally, try, catch

->Object

-> Throwable

-> Error - Error cannot be caught

-> Exception - Can be try and caught (checked) -

WashingMachineIsNotOnException

->->-> RuntimeException (unchecked exceptions) -

Superclass - WashingMachineOverLoaded

-> ArithmeticException - eg - divide by zero

Runtime exceptions - Extending from the RuntimeException class

- They need not be thrown at the method call

- The runtime exceptions need not be processed within a

try {}, catch() {} block

try and catch blocks are used for checked exceptions and not to be used
for unchecked or Runtime exceptions

finally block is used to execute a block of code regardless

File handling

Standard input device

|
|
System.in
|
|
|

Standard output device

|
|
System.out
|
|
|

+----->input----->data + functions----->output-+

|
|
Java Program

BYTE BASED STREAM - basic english

ASCII - 1 byte = 256

FileInputStream - read(); abstract class

FileOutputStream - write(); abstract class

AMIT - A=0100 0001(65), M=0100 1101(77), I=0100 1001(73), T=0101 0100(84)

> BYTE BASED representation

1111
PrintStream

CHARACTER BASED STREAM - beyond english
UNICODE - 1 char = 2 bytes = 65535
FileReader - read();
FileWriter - write();
AMIT - A=0000 0000 0100 0001(65), M=0000 0000 0100 1101(77),
2222 I=0000 0000 0100 1001(73), T=0000 0000 0101 0100(84) > CHARACTER
BASED representation
PrintWriter

type -n *.txt
vi. *.txt
type \C\Users\.....

e.printStackTrace(); - gives a complete path of the exception

-1 means end of file
Thread.sleep(no. of secs) - pauses for some amt of time specified

FileInputStream - to read a file
- Output of the file details can be done through the methods
of this class. - to print in a file
- Used to read a stream of raw bytes
- eg - FileInputStream file = new FileInputStream();
file.read(); ----> can only read as byte values (must
force convert with (byte))
file.close(); \----> must be converted to char at
the time of printing

FileOutputStream - For writing streams of raw bytes, consider using a
FileOutputStream

FileReader - The FileReader is meant for reading streams of characters

FileWriter - The FileWriter is meant for writing streams of characters

ObjectOutputStream - ObjectOutputStream writes primitive data types and
graphs of Java objects to an OutputStream

ObjectInputStream - Creates an ObjectInputStream that reads from the
specified InputStream

System.out.println();

--> out is a static object reference
--> System is the class within which the object which out refers is
present
--> println() is the method which receives String as argument, as one
among many overloaded method arguments

Object serialization process - Storing private data of the object in an external file(persistent device)

- > Only possible if the class implements - Serializable interface

- > The serializable interface has no methods, and is a marker interface

- Marker interface - It marks a contract(documentation contract)

- > static data member is never serialized

- static data members remain same for all objects of the class that are created

- > transient field is not shared among objects of the class

- > Using instanceof we can create our own marker interface