Quick Revisit of Day14

Lambda – Anonymous method used to provide implementation of Functional Interface

Lambda enables functional programming – performing multiple operations in a single line.

Functional Interface – Interface with only one abstract method (Runnable, java.util.function)

Consumer (accept(T t)), Predicate (test(T t)), Supplier (get()), Function (apply(T t))

Java.util.function package consist of interfaces only (No classes, no Exceptions, No enums defined in this package)

Lambda with zero arguments ()-> statement; or ()-> {multiple statements; … ;};

Lambda with one arguments n-> statement using n ; or (n) -> {multiple statements using n;};

Lambda with more than one argument (m,n)->statement using m&n; or (m,n)->{multiple statements using m & n; };

In Java 8, They introduced

1. Lambda
2. Functional Interface
3. @FunctionalInterface
4. Streams
5. New Date & Time, TimeZone, Locale API

IO Stream

IO – Input & Output

Stream in Java, represents flow of data.

In JAVA, there are three type of Stream.

Input Stream – Default input Stream in Java is Keyboard (System.in)

Output Stream – Default Output Stream in Java is Computer Screen / Console (System.out)

Error Stream – Default Error Stream in Java is Computer Screen/Monitor/Console (System.err)

Print() – It will print the data in a single line. If you have more than one print() method next to next, the data will be printed in a same line

Println() – prints the data in a newline.

Input (Flow of data from external world to Java program) [File/Console/Table ---🡪 Java Program]

Output (From Java Program to the External world )[Java Program -🡪 External World txt file, db table, console]

Types of Stream based on Data type

1. Byte Based Stream (Each data is of size 1 byte)
2. Character Based Stream (Each data is of size 2 bytes)

All the byte stream classes & interfaces will have stream in the end.

All the character based stream classes & interface will have Reader/Writer at the end.

IO Stream Hierarchy. ( java.io package)

InputStream & Reader classes will have read() method

OutputStream & Writer classes will have write() method

Two frequent operation with data – Read Operation & Write Operation.

Read Operation will not cause loss of data.

Write Operation will make loss of data some times. (It may over-write on existing data) – That might lead to data loss.

Streams 1) Byte based Streams 2) Character Based Streams

Two operations 1) Input Operation (read()) 2) Output Operation (write())

FileInputStream – Reading the raw data (byte) from file

FileOutputStream – Writing raw data (byte) to the file

FileReader – Reading character ( 2bytes) one by one from a File.

FileWriter – Write a character (2 bytes) to the file.

BufferedInputStream – Reading the data from a file line by line.

BufferedOutputStream – Writing the data to a file one line at a time.

File is a class- It is used to create a file or directory(folder).TO Check the given path represents a file/folder. isFile(), isFolder()

In Java, \ is a escape sequence.

Escape sequence -- \b \t – Horizontal Tab space (5 Spaces), \n – New line \r – carriage reture,

\” – To print “

\’ – To print ‘

\\ - To print \

Mac & linux operating system use / in the file path.

Serialization – It’s a process of storing the state of a object to a permanent storage.

Logging – Is the process of storing the console data to a flat file.

There are so many frameworks for logging available . (log4j, slf4j)

Log4j – Logging for JAVA

Slf4j – Simple Logging Façade (JAVA)

Logging – Recording Application Events (What is happening while running the application, when it is happening)

Each Line of Log message will consist of DATE & TIME STMAP, the class which throws the message or exception.

There are different Log levels are available

DEBUG

TRACE

WARN

INFO

ERROR

Log4j 2.17.1.jar - <https://mvnrepository.com/artifact/org.apache.logging.log4j/log4j-core/2.17.1>

This is the coding challenge for this week.

Create a Class which will display the following action items

1. Display all users
2. Add a new User
3. Update a Existing user
4. Delete a user
5. Exit

Use collection to store group of user objects. And perfrom CRUD operation on this collection.

Submit it by EOD today. 1) Create a file and copy all the code along with output screenshots.

Share this single file in email. FileName : **Week2CodingChallenge\_yourname.docx**

Sivakumar.os@revature.com

Week2 Quiz Timing – 3p.m to 5 p.m EST – Pls complete the quiz without fail.

Week3 QC will be on 2-Mar (Wednesday) 10.30 A.M to 12.30 P.M