Lecture 27: Predefined Functional Interface- Predicate Part1:

Predefined Functional Interfaces in Java

* Predicate
* Function
* Consumer
* Supplier
* Etc..

These are defined in java.util.function:

Lecture 28: Predefined Functional Interface- Predicate Part2:

Predicate:

* Predicate function:

@FunctionalInterface

public interface Predicate<T> {

boolean test(T t);

}

Eg:

**package** com.durgaSoft.section6.lecture27;

**import** java.util.function.Predicate;

**public** **class** Example1 {

**public** **static** **void** main(String[] args) {

Predicate<Integer> checkGreaterThan10 = i->i>10;

System.***out***.println(checkGreaterThan10.test(1));

System.***out***.println(checkGreaterThan10.test(12));

//We will get Incompatible type error

//System.out.println(checkGreaterThan10.test("Prashanth"));

}

}

Lecture 29: Predefined Functional Interface- Predicate Part3:

Write a predicate to check if the length of the string is greater than 5 and collection is empty or not?.

**package** com.durgaSoft.section6.lecture29;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.function.Predicate;

**public** **class** Example1 {

**public** **static** **void** main(String[] args) {

// To Check if the String length is greater than 5

Predicate<String> lengthGreaterThan5 = string -> string.length() > 5;

System.***out***.println(lengthGreaterThan5.test("JavaTest"));

System.***out***.println(lengthGreaterThan5.test("abc"));

// To check if the collectio is empty or not:

Predicate<Collection> collectionEmptyCheck =

collection -> collection.isEmpty();

ArrayList<String> arrayList1 = **new** ArrayList();

arrayList1.add("a");

System.***out***.println(collectionEmptyCheck.test(arrayList1));

ArrayList<String> arrayList2 = **new** ArrayList();

System.***out***.println(collectionEmptyCheck.test(arrayList2));

}

}

Lecture 30 Predicate Function Interface- Predicate: Joining:

* Predicate interface has the following default methods defined in the interface.
  + and
  + or
  + negate

Example:

Define an integer array and check the following for each number

* If the number is greater than 10
* If the number is even
* If the number greater than 10 and even
* If the number is less than 10 or even
* If the number is equal to 10

**int**[] number = {20,1,34,5,6,47,78,49,10};

Predicate<Integer> greaterThan10 = num -> num >10;

Predicate<Integer> evenNumber = num -> (num%2) == 0;

**for** (**int** i : number) {

System.***out***.println("Greater Than 10 "+i+" "+greaterThan10.test(i));

System.***out***.println("Even Number "+i+" "+evenNumber.test(i));

System.***out***.println(">10 and Even "+i+"

"+greaterThan10.and(evenNumber).test(i));

System.***out***.println("<10 or Even "+i+"

"+greaterThan10.negate().or(evenNumber).test(i));

}

Lecture 31:

Eg: Print out the names only if the names start with “K”

**package** com.durgaSoft.section6.lecture31;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.Iterator;

**import** java.util.function.Predicate;

**public** **class** Example1 {

**public** **static** **void** main(String[] args) {

String[] names = {"Sunny", "Kajol", "Mallika", "Katrina", "Kareena"};

Predicate<String> nameStartWithK = name -> name.startsWith("K");

**for** (String name : names) {

**if**(nameStartWithK.test(name)) {

System.***out***.println(name);

}

}

}

}

Lecture 32:

Program to print all the name of which are not null or Empty String from a given array and store the non empty names into a list.:

**package** com.durgaSoft.section6.lecture32;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.Iterator;

**import** java.util.function.Predicate;

**public** **class** Example1 {

**public** **static** **void** main(String[] args) {

String[] names = { "Durga", " ", **null**, "Ravi", "", "Shiva", **null** };

Predicate<String> toRemoveNullAndEmpty = name -> name != **null** && ! (name.trim().isEmpty());

ArrayList<String> namesList = **new** ArrayList<String>();

**for** ( String name : names) {

**if**(toRemoveNullAndEmpty.test(name)) {

System.***out***.println("Name "+name);

namesList.add(name);

}

}

System.***out***.println(namesList);

}

}

Lecture 33:

Program for user Authentication using Predicate:

**package** com.durgaSoft.section6.lecture33;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.Iterator;

**import** java.util.function.Predicate;

**public** **class** Example1 {

**public** **static** **void** main(String[] args) {

Predicate<User> userAuth = user -> user.userName == "Durga" && user.password =="Java";

User user1 = **new** User("user1", "password1");

System.***out***.println(userAuth.test(user1));

user1 = **new** User("Durga", "Java");

System.***out***.println(userAuth.test(user1));

user1 = **new** User("Durga", "password1");

System.***out***.println(userAuth.test(user1));

}

}

**class** User{

String userName;

String password;

**public** User(String userName, String password) {

// **TODO** Auto-generated constructor stub

**this**.userName = userName;

**this**.password = password;

}

}

Lecture 34: