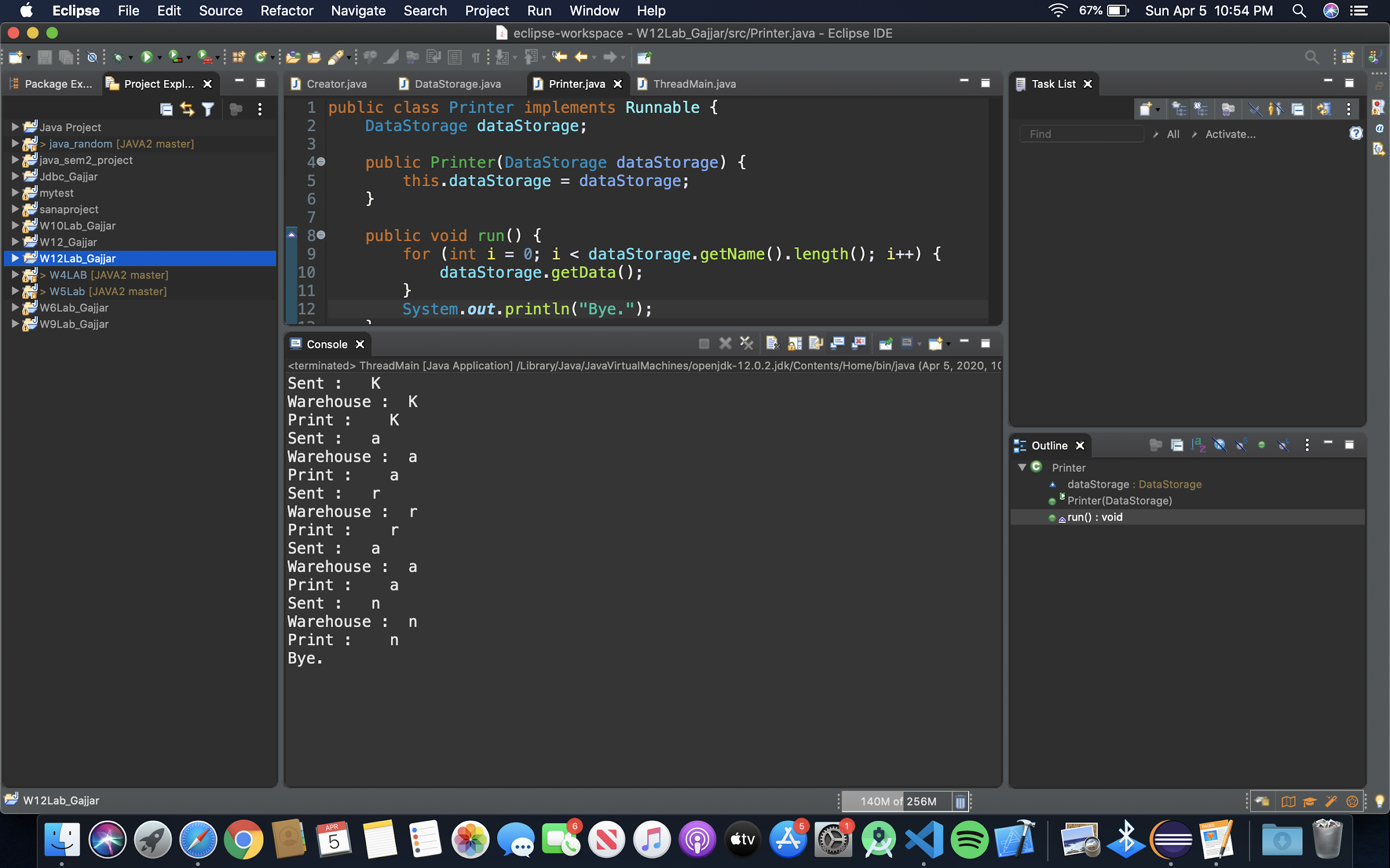
Output :



Creator :

Methods used are:

Run

Creator classed implements runnable interface which override run() method and in it using a loop each character from the name is being sent to datawarehouse and it’s sleep method allows to put a 1 second delay in which other thread print takes a character from the warehourse class array and execute it’s thread

Code:

public class Creator implements Runnable{

DataStorage dataStorage;

public Creator(DataStorage dataStorage) {

this.dataStorage = dataStorage;

}

public void run() {

char[] a = new char[dataStorage.getName().length()];

a = dataStorage.getName().toCharArray();

for(int i=0; i<dataStorage.getName().length() ; i++) {

System.***out***.println("Sent :" + " " + a[i]);

dataStorage.addData(a[i]);

try {

Thread.*sleep*(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

Datawarehouse :

This class uses synchronized method of public method which helps the other 2 thread synchronize while each character being added and read from the other two thread.

The other method used in it is wait() which causes the current thread to wait until it is notified or interrupted which addData() does.

Code :

import java.util.ArrayList;

public class DataStorage {

private String name;

private ArrayList<Character> dataWarehouselist = new ArrayList<Character>();

DataStorage(String name){

this.name=name;

}

public String getName() {

return name;

}

public synchronized void getData() {

while (dataWarehouselist.size() == 0) {

try {

wait();

} catch (InterruptedException e) {

e.printStackTrace();

}

}

System.***out***.println("Print : " + " " + dataWarehouselist.get(0));

dataWarehouselist.remove(0);

}

public synchronized void addData(char a) {

dataWarehouselist.add(a);

System.***out***.println("Warehouse :"+" "+a);

notifyAll();

}

}

Printer :

This class implements Runnable interface and run() method which gets data from warehouse list, in which getData() is called.

public class Printer implements Runnable {

DataStorage dataStorage;

public Printer(DataStorage dataStorage) {

this.dataStorage = dataStorage;

}

public void run() {

for (int i = 0; i < dataStorage.getName().length(); i++) {

dataStorage.getData();

}

System.***out***.println("Bye.");

}

}

Main :

Code :

public class ThreadMain {

public static void main(String[] args) {

String name = "Karan";

DataStorage dataWarehouse = new DataStorage(name);

Creator create = new Creator(dataWarehouse);

Printer print = new Printer(dataWarehouse);

Thread t1= new Thread(create);

t1.start();

Thread t2= new Thread(print);

t2.start();

}

}

Main method creates 2 threads and it is being started.