/\*--------------------------------------------------------

1. Name / Date: Jessica Bender / version 2 - 1/24/2021

2. Java version used, if not the official version for the class:

e.g. build 1.5.0\_06-b05:

My Java:

java version "9.0.4"

Java(TM) SE Runtime Environment (build 9.0.4+11)

Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)

3. Precise command-line compilation examples / instructions:

> javac JokeClient.java

4. Precise examples / instructions to run this program:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

5. List of files needed for running the program.

e.g.:

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

6. Notes:

For this to run since it is a little buggy put joke in JokeClientAdmin, but a username into Jokeclient then type switch to get jokes. I dont think it will give you any perverbs and it also jsut keeps running. I ran out of time and wanted to get some credit

----------------------------------------------------------\*/

// Referanced Inet homework assignment from January 2021. Some comments may overlap from JokeClient, JokeClientAdmin and JokeServer.

// Code Starts Here:

import java.io.\*;// importing all packages in java io.

import java.net.\*;// importing all packages in java net.

public class JokeClient { // start of the JokeClient class

public static void main(String[] args) {// start of main method

String serverName; //initialize serverName as a String varable.

if (args.length < 1) serverName = "localhost";//checks to see if args is less than one. If true, serverName is set equal to localhost.

else serverName = args[0];//otherwise serverName is set equal to args[0].

System.out.println("Jess Bender's Joke Client, Version 2.\n");//prints statement on terminal.

System.out.println("Using server: " + serverName + ", Port: 1581"); //prints Using server: plus whatever is saved in serverName plus the port.

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));// new BufferedReader named in

try {//trys the following code, if fails jumps to catch.

String name; //assigns a string varable to get the users username.

String another; //Assigns a string varable to get the usersinput if they want anther joke or not

int JokeNum; //Assigns a int varable to randomize the jokes

int max; //Assigns a int varable to randomize the jokes max number

int min; //Assigns a int varable to randomize the jokes min number

max = 100; //Assigning max to 100 for now not sure what the number should be yet. Maybe 4? since there are 4 Jokes and 4 Perverbs.

min = 1; //Also sasigning min to 1 for now not sure what the number should be yet. Think it should be 1 or 0?

JokeNum = (int)Math.random() \* (max - min + 1) + min; //getting a random number for jokeNum using the Math.random function. However since math.random gives a double I multiplied it by int so it would give me an int

do{//start do statement

System.out.print("Welcome to the JokeServer! Enter a username to get your first Joke or Proverb or type 'quit' to exit program: ");//prints statement on terminal. Saying welcome and Asking for the users username

System.out.flush ();

name = in.readLine ();//assigns the text from the BufferedReader in to name.

if(name.length()>=1){

System.out.print("Hello "+ name + ",\nLet's get your first Joke!\n");//prints statement on terminal. Saying welcome

}else if(name.length()<1){

System.out.print("Im sorry I did not catch your name.\n");//prints statement on terminal. Saying ne name was entered

}else{

System.out.println ("Exiting program. Come back Soon!");//when another = end print this statement. saying they are exiting the program and tell them to come back soon!

}

}//close do

while (name.indexOf("quit") < 0 && name.length()<1);// keep doing the do above until another = end

do {//Start 2nd do statement

System.out.print("To receive your next Joke or Proverb enter 'next'! To swich from Joke to Proverb or vise versa enter 'switch'! To end program type 'end': ");//prints statement on terminal. Asking the user if they want another joke or not

System.out.flush ();

another = in.readLine ();//assigns the text from the BufferedReader in to another.

if (another.indexOf("end") < 0 && another.equals("switch")){//checks to see if another = end and switch

getRemoteAddress(another, serverName);//calls function getRemoteAddress below and puts in name and serverName for the second string varables.

}//close if

}

//closes 2nd do

while (another.indexOf("end") < 0);// keep doing the do above until another = end

System.out.println ("Exiting program. Come back for more Jokes and Perverbs soon!");//when another = end print this statement. saying they are exiting the program and tell them to come back soon!

}//closes try

catch (IOException x) {x.printStackTrace ();} //catches IOExeption when try fails and prints the error.

}//closes main

//not sure if I need the toText methond from Inet will leave here for now tho

static String toText (byte ip[]) { //new method

StringBuffer result = new StringBuffer (); //new StringBuffer called result.

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

if (i > 0) result.append ("."); // add . to result if i is larger than 0.

result.append (0xff & ip[i]); // if not greater than 0 0xff & ip[i] gets added to result.

}//closes for

return result.toString ();// returns result as a string

}//closes toText

static void getRemoteAddress (String name, String serverName){ //// gets it name and serverName string variables and JokeNum int varable from above in the do statement

Socket sock; //makes Socket variable called sock.

BufferedReader fromServer; //makes BufferedReader variable called fromServer.

PrintStream toServer; //makes PrintStream variable called toServer.

String textFromServer; //makes String variable called textFromServer.

try{//trys the following code, if fails jumps to catch.

sock = new Socket(serverName, 1591); //assigns sock toa new Socket with serverName and the port 1581

fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream())); //assigns fromServer to a new BufferedReader that gets the input from sock

toServer = new PrintStream(sock.getOutputStream()); //assigns toServer to a new PrintStream that writes the input from sock

toServer.println(name); toServer.flush(); //Prints the name and the JokeNum

for (int i = 1; i <=3; i++){

textFromServer = fromServer.readLine();//assigns textFromServer to the text in fromServer.

if (textFromServer != null) System.out.println(textFromServer); //checks to make sure textFromServer is not null/ empty and id so prints whatever was in textFromServer.

}//closes for

sock.close();//closes sock

} //closes try

catch (IOException x) { //catches IOExeption when try fails.

System.out.println ("Socket error.");

x.printStackTrace ();//prints Socket error and printStackTrace witch prints details about the error including the line number where the error occurred.

}//closes catch

}//closes getRemoteAddress

}// End of JokeClient class

/\*--------------------------------------------------------

1. Name / Date: Jessica Bender / version 2 - 1/24/2021

2. Java version used, if not the official version for the class:

My Java:

java version "9.0.4"

Java(TM) SE Runtime Environment (build 9.0.4+11)

Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)

3. Precise command-line compilation examples / instructions:

> javac JokeClientAdmin.java

4. Precise examples / instructions to run this program:

e.g.:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

All acceptable commands are displayed on the various consoles.

This runs across machines, in which case you have to pass the IP address of

the server to the clients. For exmaple, if the server is running at

140.192.1.22 then you would type:

> java JokeClient 140.192.1.22

> java JokeClientAdmin 140.192.1.22

5. List of files needed for running the program.

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

5. Notes:

For this to run since it is a little buggy put joke in JokeClientAdmin, but a username into Jokeclient then type switch to get jokes. I dont think it will give you any perverbs and it also jsut keeps running. I ran out of time and wanted to get some credit

----------------------------------------------------------\*/

// Referanced Inet homework assignment from January 2021. Some comments may overlap from JokeClient, JokeClientAdmin and JokeServer.

// Code Starts Here:

import java.io.\*;// importing all packages in java io.

import java.net.\*;// importing all packages in java net.

public class JokeClientAdmin { // start of the JokeClientAdmin class

public static void main(String[] args) {

String serverName; //initialize serverName as a String varable.

if (args.length < 1) serverName = "localhost";//checks to see if args is less than one. If true, serverName is set equal to localhost.

else serverName = args[0];//otherwise serverName is set equal to args[0].

System.out.println("Jess Bender's Joke Client Admin, Version 2.\n");//prints statement on terminal.

System.out.println("Using server: " + serverName + ", Port: 1881"); //prints Using server: plus whatever is saved in serverName plus the port.

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));// new BufferedReader named in

try {//trys the following code, if fails jumps to catch.

String or;

do {

System.out.print("This is the Admin Client, Please enter weather you want a 'joke' or 'perverb' or 'quit' to end program: ");//prints statement on terminal.

System.out.flush ();

or = in.readLine ();//assigns the text from the BufferedReader in to or.

if (or.indexOf("quit") < 0)//checks to see if or = quit

getRemoteAddress(or, serverName);//calls function getRemoteAddress below and puts in or and serverName for the 2 string varables.

}//closes do

while (or.indexOf("quit") < 0);// keep doing the do above until or = quit

System.out.println ("Leaving Program......GoodBye!");//when or = quit print this statement.

}//closes try

catch (IOException x) {x.printStackTrace ();} //catches IOExeption when try fails and prints the error.

}//closes main

static void getRemoteAddress (String or, String serverName){ //// gets it or and serverName variables from above in the do statement

Socket sock; //makes Socket variable called sock.

BufferedReader fromServer; //makes BufferedReader variable called fromServer.

PrintStream toServer; //makes PrintStream variable called toServer.

try{//trys the following code, if fails jumps to catch.

sock = new Socket(serverName, 2571); //assigns sock toa new Socket with serverName and the port 1881

fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream())); //assigns fromServer to a new BufferedReader that gets the input from sock

toServer = new PrintStream(sock.getOutputStream()); //assigns toServer to a new PrintStream that writes the input from sock

toServer.println(or); toServer.flush();

System.out.println(fromServer.readLine()); //checks to make sure textFromServer is not null/ empty and id so prints whatever was in textFromServer.

} //closes try

catch (IOException x) { //catches IOExeption when try fails.

System.out.println ("Socket error.");

x.printStackTrace ();//prints Socket error and printStackTrace witch prints details about the error including the line number where the error occurred.

}//closes catch

}//closes getRemoteAddress

}// End of JokeClientAdmin class

/\*--------------------------------------------------------

1. Name / Date: Jessica Bender / Version 2 1/24/2021

2. Java version used, if not the official version for the class:

My Java:

java version "9.0.4"

Java(TM) SE Runtime Environment (build 9.0.4+11)

Java HotSpot(TM) 64-Bit Server VM (build 9.0.4+11, mixed mode)

3. Precise command-line compilation examples / instructions:

> javac JokeServer.java

4. Precise examples / instructions to run this program:

In separate shell windows:

> java JokeServer

> java JokeClient

> java JokeClientAdmin

5. List of files needed for running the program.

e.g.:

a. checklist.html

b. JokeServer.java

c. JokeClient.java

d. JokeClientAdmin.java

6. Notes:

For this to run since it is a little buggy put joke in JokeClientAdmin, but a username into Jokeclient then type switch to get jokes. I dont think it will give you any perverbs and it also jsut keeps running. I ran out of time and wanted to get some credit

----------------------------------------------------------\*/

// Referanced Inet homework assignment from January 2021. Some comments may overlap from JokeClient, JokeClientAdmin and JokeServer.

// All Jokes were taken from google. here is the link used https://www.rd.com/list/short-jokes/

// All Proverbs were taken from google. here is the link used https://lemongrad.com/proverbs-with-meanings-and-examples/

// Code Starts Here:

import java.io.\*;// importing all packages in java io.

import java.net.\*;// importing all packages in java net.

import java.util.HashMap; //importing hashmaps to use to store jokes and proverbs

class Worker extends Thread{ //start Worker class. This class will

Socket sock; //makes a new Socket called sock.

Worker (Socket s) {sock = s;}

public void run(){

PrintStream out = null; //assigns a PrintStream variable called out to null. PrintStream is a directory in the java.io package that writes output data.

BufferedReader in = null; //assigns a BufferedReader variable called in to null. BufferedReader is a directory in the java.io package that reads text from an input including sockets

try { //will try the things within the try otherwise will skip to catch

in = new BufferedReader // assigns in to a new BufferedReader

(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream()); // assigns out to a new PrintStream that gets its data from the Socket sock .getInputStream() allows us to get that output

try { //will try the things within the try otherwise will skip to catch

String name;//makes a new string called name

name = in.readLine ();//assigns the string variable name to what was read in the the varaible in.

if(JokeServer.joke == true){

System.out.println("Looking up " + name + "Joke"); //prints Lokking up and whatever was saved into the variable name to the terminal.

printJoke(name, out); //calls the static void printJoke a few lines down and sets the String to whatever the name vaible was assigned to and the PrintStream to whatever the out vaible was assigned to.

}//close if

else if(JokeServer.joke == false){

System.out.println("Looking up " + name + "Proverb"); //prints Lokking up and whatever was saved into the variable name to the terminal.

printProverb(name, out); //calls the static void printJoke a few lines down and sets the String to whatever the name vaible was assigned to and the PrintStream to whatever the out vaible was assigned to.

}//close if

}//closes second try

catch (IOException x) { //catches IOExeption when try fails

System.out.println("Server read error"); x.printStackTrace (); //prints Server read error and printStackTrace witch prints details about the error including the line number where the error occurred.

} //closes catch

sock.close(); // closes the Socket called sock

} //closes first try

catch (IOException ioe) {System.out.println(ioe);} //catches IOExeption when try fails and prints the error

} // closes run()

static void printJoke(String name, PrintStream out){//Start printJoke method

while(true){

int random = (int)(Math.random() \* 4) + 1;//generates a random number from 1 to 5

int[] usedjokes = new int[] {0,0,0,0};//int array to see if joke was used 0 = not used 1 = used

if (random == 1 && usedjokes[0] == 0){

usedjokes[0]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JA"));//printing joke A

}//end if

else if(random == 2 && usedjokes[1] == 0){

usedjokes[1]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JB"));//printing joke B

}//clsoes else if

else if(random == 3 && usedjokes[2] == 0){

usedjokes[2]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JC"));//printing joke B

}//clsoes else if

else if(random == 4 && usedjokes[3] == 0){

usedjokes[3]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JD"));//printing joke B

}//clsoes else if

}//end while

}//end printJoke method

static void printProverb(String name, PrintStream out){//Start printProverb method

while(true){

int[] usedProvervs = new int[] {0,0,0,0};//int array to see if proverb was used 0 = not used 1 = used

int random = (int)(Math.random() \* 5) + 1;//generates a random number from 1 to 5

if (random == 1 && usedProvervs[0] == 0){

usedProvervs[0]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JA"));//printing joke A

}//end if

else if(random == 2 && usedProvervs[1] == 0){

usedProvervs[1]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JB"));//printing joke B

}//clsoes else if

else if(random == 3 && usedProvervs[2] == 0){

usedProvervs[2]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JC"));//printing joke B

}//clsoes else if

else if(random == 4 && usedProvervs[3] == 0){

usedProvervs[3]=1;//setting it to 1 now so i know it was used

System.out.println(JokeServer.joke("JD"));//printing joke B

}//clsoes else if

}//end while

}//end printProverb method

}//closes worker class

//AdminLooper class given by class website

class AdminLooper implements Runnable {

public static boolean adminControlSwitch = true;

public void run(){ //runs

System.out.println("In the admin looper thread");//prints to termail

int q\_len = 6; //q\_len is = 6

int port = 2571; // new port number for admin

Socket sock;//new socet called soc

try{//try

ServerSocket servsock = new ServerSocket(port, q\_len); //New ServerSocket Called sevsock with port and len

while (adminControlSwitch) { //while

sock = servsock.accept();

new AdminWorker (sock).start();

}//close while

}//close try

catch (IOException ioe) {System.out.println(ioe);}//catches when try fails

}//closes run

}//closes AdminLooper

class AdminWorker extends Thread{//begining half is same as worker

Socket sock;//new socet called soc

AdminWorker (Socket s) {sock = s;}

public void run(){

PrintStream out = null; //assigns a PrintStream variable called out to null. PrintStream is a directory in the java.io package that writes output data.

BufferedReader in = null; //assigns a BufferedReader variable called in to null. BufferedReader is a directory in the java.io package that reads text from an input including sockets

try { //will try the things within the try otherwise will skip to catch

in = new BufferedReader // assigns in to a new BufferedReader

(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream()); // assigns out to a new PrintStream that gets its data from the Socket sock .getInputStream() allows us to get that output

try { //will try the things within the try otherwise will skip to catch

String choice;//makes a new string called choice

choice = in.readLine ();//assigns the string variable choice to what was read in the the varaible in.

if (choice.indexOf("joke") < 0){//if joke=true

JokeServer.joke = true;

JokeServer.proverb = false;

System.out.println("Let's Get some Jokes!");

out.println("Let's Get some Jokes!");

}//closes if

else{

JokeServer.proverb = true;

JokeServer.joke = false;

System.out.println("Let's Get some Proverbs!");

out.println("Let's Get some Proverbs!");

}

}//closes try

catch (IOException ioe) {System.out.println(ioe);}//catches when try fails

}//closes try

catch (IOException ioe) {System.out.println(ioe);}//catches when try fails

}//closes run

}//closes AdminWorker

public class JokeServer { // start of the JokeServer class

static String choice ="proverb";//makes a static varable to see if the user is on joke or proverb. I am also initizing it as a proverb

static Boolean joke; //makes a static boolean for joke

static Boolean proverb; //makes a static boolean for proverb

public static String joke(String haha){ //start of joke method take in a string haha

HashMap<String, String> Jokes = new HashMap<String, String>(); //making a new hashmap to store my jokes. First string in the hashmap will be for the joke numbers ie JA, JB, JC, and JD. the second string will be the actual joke.

Jokes.put("JA", "Why did the nurse need a red pen at work? In case she needed to draw blood."); //first joke.

Jokes.put("JB", "I invented a new word! Plagiarism!"); //Second joke.

Jokes.put("JC", "Yesterday I saw a guy spill all his scrabble letter on the road. I asked him, 'what's the word on the street?'"); //Third joke.

Jokes.put("JD", "Hear about the new resturant called Karma? There's no menu: You get what you deserve."); //Fouth joke.

return haha+":" + Jokes.get(haha);

}//closes joke

public static String proverb(String wisdom){ //start of joke method take in a string haha

HashMap<String, String> Proverbs = new HashMap<String, String>(); //making a new hashmap to store my jokes. First string in the hashmap will be for the joke numbers ie JA, JB, JC, and JD. the second string will be the actual joke.

Proverbs.put("PA", "A bad workman always blames his tools."); //first Proverb.

Proverbs.put("PB", "A journey of thousand miles begins with a single step."); //Second Proverb.

Proverbs.put("PC", "Beggars canâ€™t be choosers."); //Third Proverb.

Proverbs.put("PD", "Beauty is in the eye of the beholder."); //Fouth Proverb.

return wisdom+":" + Proverbs.get(wisdom);

}//closes proverb

public static void main(String a[]) throws IOException {// start of main method

int q\_len = 6; //initialize q\_len as an int and assigns it to 6

int port = 1591;//initialize port as an int and assigns a port number(can be changed)

Socket sock; //makes a Socket called sock

AdminLooper loop = new AdminLooper();//new Adminlooper called loop

Thread thread = new Thread(loop);//new tread taking in AdminLooper

thread.start();

ServerSocket servsock = new ServerSocket(port, q\_len); //makes a new ServerSocket called servsock and puts the port number and q\_len assigned above.

System.out.println("Jess Bender's Joke Server, Version 2.\n");//prints statement on terminal.

System.out.println("Listening at port:"+port+ "\n");//prints that text on the termial

while (true) { //keeps running while its true

sock = servsock.accept(); //assignes sock to an accepted servsock.

new Worker(sock).start(); //calls on worker class with the sock assigned in it.

} //closes while

}// end of main method

}// End of JokeServer class