import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.text.ParseException;

/\*\*

\* Created by Hemanth on 9/23/16.

\*/

public class AdminMode extends Thread {

private Status status;

private Socket sock;

AdminMode(Socket sock, Status status) {

this.status = status;

this.sock = sock;

}

@Override

public void run() {

System.out.println("Admin Connected");

PrintStream out = null;

BufferedReader in = null;

try {

in = new BufferedReader(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream());

try {

String name;

name = in.readLine();

System.out.println("Request From Admin: " + name);

try {

evaluteRequest(Integer.parseInt(name),out);

//Send back that the input is not an Number

}catch (NumberFormatException ioe){out.println("Invalid Numerical Choice");}

} catch (IOException ioe) {

System.out.println("Server Read Error");

}

sock.close();

} catch (IOException x) {

System.out.println(x);

}

}

/\*\*

\* Chooses correct path based on user input

\* @param check

\* @param output

\*/

public void evaluteRequest(int check,PrintStream output){

switch(check){

case 0:

System.out.println("Admin Selection Change Server State");

status.setJoke();

output.println("Current Status of Joke State: " + status.toString());

break;

case 1:

System.out.println("Admin: Send Server State " + status.toString());

output.println(status.toString());

break;

case 2:

System.out.println("Admin: Shutdown Initiated");

output.println("Shutdown Initiated");

System.exit(0);

break;

default :

System.out.println("Admin: Not a valid option");

output.println("Not a valid option");

}

}

}import java.net.ServerSocket;

import java.net.Socket;

/\*\*

\* Created by Hemanth on 9/23/16.

\*/

public class AdminServer extends Thread {

private static Socket sock;

private Status status;

private int port;

AdminServer(Status status,int port){this.status = status;this.port = port;}

@Override

public void run(){

int q\_len = 6;

try{

ServerSocket adminsock = new ServerSocket(port,q\_len);

System.out.println("Starting Admin Server PORT:" + port);

//Server Forever listing to port for Admin Connection

while (true) {

sock = adminsock.accept();

//

//When Admin connects handle it

new AdminMode (sock,status).start();

}

}catch (Exception e){System.out.println(e);}

}

}import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.text.ParseException;

/\*\*

\* Created by Hemanth on 9/23/16.

\*/

public class AdminMode extends Thread {

private Status status;

private Socket sock;

AdminMode(Socket sock, Status status) {

this.status = status;

this.sock = sock;

}

@Override

public void run() {

System.out.println("Admin Connected");

PrintStream out = null;

BufferedReader in = null;

try {

in = new BufferedReader(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream());

try {

String name;

name = in.readLine();

System.out.println("Request From Admin: " + name);

try {

evaluteRequest(Integer.parseInt(name),out);

//Send back that the input is not an Number

}catch (NumberFormatException ioe){out.println("Invalid Numerical Choice");}

} catch (IOException ioe) {

System.out.println("Server Read Error");

}

sock.close();

} catch (IOException x) {

System.out.println(x);

}

}

/\*\*

\* Chooses correct path based on user input

\* @param check

\* @param output

\*/

public void evaluteRequest(int check,PrintStream output){

switch(check){

case 0:

System.out.println("Admin Selection Change Server State");

status.setJoke();

output.println("Current Status of Joke State: " + status.toString());

break;

case 1:

System.out.println("Admin: Send Server State " + status.toString());

output.println(status.toString());

break;

case 2:

System.out.println("Admin: Shutdown Initiated");

output.println("Shutdown Initiated");

System.exit(0);

break;

default :

System.out.println("Admin: Not a valid option");

output.println("Not a valid option");

}

}

}import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.util.Scanner;

import java.util.UUID;

/\*\*

\* Created by Hemanth on 9/10/2016.

\*/

public class JokeClient {

private static String uniqueID = UUID.randomUUID().toString();

private static String user = "Default";

private static int num = 0;

private static String ip1 = null;

private static String ip2 = null;

private static String serverName = "localhost";

private static int port = 4545;

public static void main(String args[]) {

// if (args.length < 2) {

// serverName = "localhost";

// port = 4545;

// System.out.println("Not using 2 Commandline arguments for servername and port using defaults");

// } else {

// serverName = args[0];

// port = Integer.parseInt(args[1]);

// }

setIP(args);

System.out.println("Hemanth Ande's Joke Client, 1.8.");

System.out.println("Using Server: " + serverName + ", Port: " + port);

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

Scanner user\_input = new Scanner(System.in);

System.out.print("Enter your name: ");

user = user\_input.next();

System.out.println("Welcome " + user + "!");

try {

String name;

do {

System.out.println("Press Enter to receive Joke/proverb,(ip1,ip2) to swtich Servers (quit) to end: ");

System.out.flush();

name = in.readLine();

if (name.toLowerCase().equals("ip1")) {

if (ip1 != (null)) {

System.out.println("Changing to " + ip1);

serverName = ip1;

port = 4545;

}else{

System.out.println("Multiple Ips Not Found");

}

} else if (name.toLowerCase().equals("ip2")) {

if (ip2 != (null)) {

System.out.println("Changing to " + ip2);

serverName = ip2;

port = 4546;

}else{

System.out.println("Multiple Ips Not Found");

}

}

//If quit substring in name stop

if (!name.toLowerCase().equals("quit") && name.equals("")) {

//System.out.println("Sending\n"+ name);

sendMessage(uniqueID, serverName, port);

num++;

}

//After 5 excutes print for formating reasons (5 jokes and proverbs)

if (num % 5 == 0) {

System.out.println("=============================================");

}

} while (!name.toLowerCase().equals("quit"));

System.out.println("Cancelled by user request.");

} catch (IOException ioe) {

ioe.printStackTrace();

}

}

/\*\*

\* Deprecated not in use

\* Converts byte to string

\*

\* @param ip byte array to be converted to string

\* @return result String Converted IP addres

\*/

static String toText(byte ip[]) {

StringBuffer result = new StringBuffer();

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

if (i > 0) result.append(".");

result.append(0xff & ip[i]);

}

return result.toString();

}

/\*\*

\* Sends input name to Server receives name address of input

\*

\* @param name String name of website

\* @param serverName IP address of website

\*/

static void sendMessage(String name, String serverName, int port) {

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

String textFromServer;

try {

sock = new Socket(serverName, port);

fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream()));

toServer = new PrintStream(sock.getOutputStream());

name = name + ":" + user;

//Send name to server and then clear

toServer.println(name);

toServer.flush();

//Reads blocks of data from the server (up to 3 lines)

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

textFromServer = fromServer.readLine();

if (textFromServer != null) System.out.println(textFromServer);

}

sock.close();

} catch (IOException ioe) {

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

ioe.printStackTrace();

}

}

/\*\*

\* Setups env for dual connections to servers

\* @param args

\*/

public static void setIP(String args[]) {

if (args.length == 2) {

System.out.println("Reading Commandline Args");

ip1 = args[0];

ip2 = args[1];

}if (args.length == 1){

System.out.println("Found Custom IP " + args[0]);

serverName = args[0];

}

else{

System.out.println("2 Commandline Args Not Found");

}

}

}<------------------------------------------------------------------------------------------------------------------------------------------------------------------->

//10 response in Joke mode captured from JokeClient

Not using 2 Commandline arguments for servername and port using defaults

Hemanth Ande's inet Client, 1.8.

Using Server: localhost, Port: 4545

Enter your name: player

Welcome player!

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JC player : When I call a family meeting I turn off the house wifi and wait for them all to come running.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JE player : I recently decided to sell my vacuum cleaner as all it was doing was gathering dust.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JB player : Today a man knocked on my door and asked for a small donation towards the local swimming pool. I gave him a glass of water.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JA player : Life is all about perspective. The sinking of the Titanic was a miracle to the lobsters in the ship's kitchen.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JD player : I refused to believe my road worker father was stealing from his job, but when I got home all the signs were there.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JE player : I recently decided to sell my vacuum cleaner as all it was doing was gathering dust.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JA player : Life is all about perspective. The sinking of the Titanic was a miracle to the lobsters in the ship's kitchen.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JD player : I refused to believe my road worker father was stealing from his job, but when I got home all the signs were there.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JC player : When I call a family meeting I turn off the house wifi and wait for them all to come running.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JB player : Today a man knocked on my door and asked for a small donation towards the local swimming pool. I gave him a glass of water.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

<------------------------------------------------------------------------------------------------------------------------------------------------------------------->

//10 Responses in Proverb Mode from JokeClient

Not using 2 Commandline arguments for servername and port using defaults

Hemanth Ande's inet Client, 1.8.

Using Server: localhost, Port: 4545

Enter your name: player

Welcome player!

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PC player : The pursuit of happiness is the chase of a life time.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PA player : Those who say it can't be done are usually interrupted by others doing it.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PB player : A smile is an inexpensive way to change your looks.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PE player : Better to understand little than to misunderstand a lot.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PD player : Every man dies; but not every man really lives.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PE player : Better to understand little than to misunderstand a lot.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PC player : The pursuit of happiness is the chase of a life time.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PD player : Every man dies; but not every man really lives.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PA player : Those who say it can't be done are usually interrupted by others doing it.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PB player : A smile is an inexpensive way to change your looks.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

<------------------------------------------------------------------------------------------------------------------------------------------------------------------->

//Interleaving of modes

Not using 2 Commandline arguments for servername and port using defaults

Hemanth Ande's inet Client, 1.8.

Using Server: localhost, Port: 4545

Enter your name: Player

Welcome Player!

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JC Player : When I call a family meeting I turn off the house wifi and wait for them all to come running.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PC Player : The pursuit of happiness is the chase of a life time.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JD Player : I refused to believe my road worker father was stealing from his job, but when I got home all the signs were there.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PB Player : A smile is an inexpensive way to change your looks.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JB Player : Today a man knocked on my door and asked for a small donation towards the local swimming pool. I gave him a glass of water.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PD Player : Every man dies; but not every man really lives.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JE Player : I recently decided to sell my vacuum cleaner as all it was doing was gathering dust.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PE Player : Better to understand little than to misunderstand a lot.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> JA Player : Life is all about perspective. The sinking of the Titanic was a miracle to the lobsters in the ship's kitchen.

Press Enter to receive Joke/proverb, (quit) to end:

<s1> PA Player : Those who say it can't be done are usually interrupted by others doing it.

=============================================

Press Enter to receive Joke/proverb, (quit) to end:

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.util.Scanner;

/\*\*

\* Created by Hemanth on 9/23/16.

\*/

public class JokeClientAdmin {

private static String ip1;

private static String ip2;

private static String serverName = "localhost";

private static int port = 5050;

public static void main (String args[]){

if (args.length>0) {

if (checkArgs(args)){

System.out.println("Found Custom Args");

}else {

serverName = "localhost";

port = 5050;

System.out.println("Admin Using Defaults");

}

}

System.out.println("Admin Client, 1.8.");

System.out.println("Connecting to Server: " + serverName + ", Port: " + port);

String state = "Not Known";

setIP(args);

Socket sock;

try{

String name;

do{

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

System.out.println("Please select option, type (quit) to exit");

System.out.println("Last Known Server State: " + state);

//

printOptions("Toogle Server Status",0);

printOptions("Server State Status", 1);

printOptions("Shutdown Server",2);

printOptions("Change Server Connection",3);

System.out.println("Please Enter Selection: ");

System.out.flush();

name = in.readLine();

//If not 'quit' send the message

if (name.toLowerCase().equals("3")){

if (!ip1.equals(null) || !ip2.equals(null)){

if (serverName.equals(ip1)){

serverName = ip1;

port = 5050;

}else {

serverName = ip2;

port = 5151;

}

}else{

System.out.println("Unable to Change no second ip");

}

System.out.println("Changing to " + serverName);

}

if (!name.toLowerCase().equals("quit")){

//System.out.println("Sending\n"+ name);

String check = sendMessage(name,serverName,port);

if (!check.equals("False")) {

//Get Server State every send message

state = currentServerState("1", serverName, port);

}else{

//Quit Program if unable to communicate with server

System.exit(21);

}

}

//If quit is found stop program

}while (!name.toLowerCase().equals("quit"));

System.out.println("Cancelled by user request.");

}catch (IOException ioe){System.out.println("Unable to Connect to Server");}

}

/\*\*

Receives String to send,ip address and port

Returns String from server

\*/

static String sendMessage(String name , String serverName, int port){

Socket sock;

BufferedReader fromServer;

PrintStream toServer;

//Default Response if we cannot connect to server

String textFromServer = "No Reply";

try{

sock = new Socket (serverName,port);

fromServer = new BufferedReader(new InputStreamReader(sock.getInputStream()));

toServer = new PrintStream(sock.getOutputStream());

//name = name ;

//Send name to server and then clear

toServer.println(name); toServer.flush();

//Reads 3 lines of data one at a time

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

textFromServer = fromServer.readLine();

//break since we are return the string below

if (textFromServer != null )break;

}

sock.close();

}catch (IOException ioe){System.out.println("Unable to communicate with server");return "False";}

return textFromServer;

}

/\*\*

Pretty print of options

\*/

static void printOptions(String option, int optionNo){

System.out.println("[" + optionNo +"] " + option );

}

/\*\*

Abstracted sendMessage functionality into currentServerState for more custom return String

\*/

static String currentServerState(String choice, String serverName, int port){

try{

return sendMessage(choice,serverName, port);

//System.out.println("Received State " + hello );

//return hello;

}catch (Exception ioe){return "Not Known";}

}

/\*\*

\* Checks args to make correct number if found

\* @param args

\* @return

\*/

public static boolean checkArgs(String args[]){

if (args.length == 2){

ip1 = args[0];

ip2 = args[1];

System.out.println("Found multiple servers");

selectIP();

}else if (args.length ==1){

ip1 = args[0];

}else{

System.out.println("Max Args is two");

return false;

}

return true;

}

/\*\*

\* Allows user to select IP to connect to

\*/

public static void selectIP(){

System.out.println("Select Server to connect to");

System.out.println("[0] " + ip1);

System.out.println("[1] " + ip2);

try{

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

String enter = in.readLine();

switch (enter){

case "0":

serverName = ip1;

port = 5050;

break;

case "1":

serverName = ip2;

port = 5051;

break;

default:

System.out.println("Option not valid");

selectIP();

}

}catch (IOException ioe){System.out.println("Unable to read input");}

}

/\*\*

\* Sets custom or mutiple ips for server

\* @param args

\*/

public static void setIP(String args[]){

if (args.length == 2){

System.out.println("Reading Commandline Arguments");

ip1 = args[0];

ip2 = args[1];

}else if (args.length == 1){

System.out.println("Found One Commandline Arg");

serverName = args[0];

}

else{

System.out.println("Unable TO Find two Commandline Args");

}

}

}

import java.io.IOException;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.HashMap;

/\*\*

\* Created by Hemanth on 9/25/16.

\*/

public class secondaryServer extends Thread {

private boolean Joke = true;

static private HashMap<String, Integer> uidList = new HashMap<String, Integer>();

static private HashMap<String, Integer> uidCycle = new HashMap<String, Integer>();

private static Status status ;

private static Boolean secondary = false;

private static int port;

public secondaryServer(HashMap<String, Integer>uidList, Status status, HashMap<String,Integer>uidCycle, Boolean secondary,int port){

this.uidCycle = uidCycle;

this.uidList = uidList;

this.status = status;

this.secondary = secondary;

this.port = port;

}

@Override

public void run(){

//System.out.println("Secondary Server Started");

try {

Socket sock;

int q\_len = 6;

ServerSocket servsock = new ServerSocket(port, q\_len);

System.out.println("Hemanth Ande Second Joke Server is running on port " + port);

while (true) {

try {

sock = servsock.accept();

System.out.println("received Connection");

//sock2 = adminsock.accept();

//System.out.println("Current Joke Staus: " + status.toString());

new Worker(sock, uidList, status, uidCycle,secondary).start();

} catch (IOException ioe) {

System.out.println("Unable to start server");

}

}

} catch (IOException ioe) {

System.out.println("Unable to start second server");

System.out.println(ioe.toString());

}

}

}

/\*\*

\* Created by Hemanth on 9/23/16.

\*/

public class Status extends Thread {

static Boolean status = true;

@Override

public void run (){

System.out.println("Current Joke Status" + printJoke());

while (true) {

}

}

/\*\*

\* Toogle for current mode

\*/

public void setJoke(){

status = !status;

System.out.println(printJoke());

}

/\*\*

\*

\* @return Boolean State

\*/

public Boolean getJoke(){

return status;

}

/\*\*

\*

\* @return Pretty print of current State

\*/

public String printJoke(){

if (status){

return "Current Joke State: Joke";

}else{

return "Current Joke State: Proverb";

}

}

/\*\*

\* rewrote to get string of the boolean value

\* @return

\*/

@Override

public String toString() {

if (status){

return "Joke";

}else{

return "Proverb";

}

}

}

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintStream;

import java.net.Socket;

import java.util.\*;

/\*\*

\* Created by Hemanth on 9/18/2016.

\*/

public class Worker extends Thread {

Socket sock;

Worker(Socket s,HashMap<String,Integer> al,Status status,HashMap<String,Integer> wl,Boolean secondary ) {sock =s;collect=al;Joke = status; uidCycle = wl;this.secondary = secondary; }

HashMap<String,Integer> collect = new HashMap<String,Integer>();

HashMap<String,Integer> uidCycle = new HashMap<String,Integer>();

Status Joke;

Boolean secondary;

public void run(){

PrintStream out = null;

BufferedReader in = null;

try{

in = new BufferedReader(new InputStreamReader(sock.getInputStream()));

out = new PrintStream(sock.getOutputStream());

try{

String name;

name = in.readLine();

//System.out.println("Request From: " + name);

//System.out.println("Sending value "+name+Joke.getJoke());

//collect.put(name+Joke.toString(),0);

printList(collect);

//out.println("Send things over\n");

//setJoke();

//out.println(null);

if (checkNew(name+Joke.getJoke())) {

String data = getData(Joke, name);

out.print(data);

}

}catch (IOException ioe){System.out.println("Server Read Error");}

sock.close();

}catch (IOException x){

System.out.println(x);

}

}

/\*\*

\* Toggles between Joke and Proverb

\*/

// public void setJoke(){

// //Boolean local\_value = JokeServer.STATUS;

// JokeServer.setJoke();

// }

/\*\*

\* For troublshooting purposes print current list

\* @param al

\*/

public void printList(HashMap<String,Integer> al){

for (String key : al.keySet()){

System.out.println(key + " : " + al.get(key));

}

}

/\*\*

\* Chooser between getting joke or proverb

\* @param status

\* @param UUID

\* @return data

\*/

public String getData(Status status,String UUID){

if (status.getJoke()){

System.out.println("Getting Joke");

return getJoke(UUID);

}else{

System.out.println("Getting Proverb");

return getProverb(UUID);

}

}

//Jokes are from onelinefun.com

/\*\*

\* Chooses the correct joke from random seed

\* @param UUID

\* @return Sends Joke to client

\*/

public String getJoke(String UUID){

String dataString;

String unique = UUID+"true";

int selection = collect.get(unique);

if (selection == 5){

collect.put((UUID+"true"),0);

selection = 0;

}

List<Integer> messages = Arrays.asList(0, 1, 2, 3, 4);

messages = randomSeed(messages,(UUID+"true"));

//Collections.shuffle(messages, new Random(unique.hashCode()));

//System.out.println(messages);

String user = UUID.split(":")[1];

System.out.println("Current UUID: " + UUID + " Joke Selection : " + selection);

selection = messages.get(selection);

String serverTag = serverTagId(secondary);

String defaultString = serverTag + "JA " + user + " : Life is all about perspective. The sinking of the Titanic was a miracle to the lobsters in the ship's kitchen.";

switch (selection){

case 0:

dataString = defaultString;

break;

case 1:

dataString = serverTag + "JB " + user + " : Today a man knocked on my door and asked for a small donation towards the local swimming pool. I gave him a glass of water.";

break;

case 2:

dataString = serverTag + "JC "+ user + " : When I call a family meeting I turn off the house wifi and wait for them all to come running.";

break;

case 3:

dataString = serverTag + "JD " + user + " : I refused to believe my road worker father was stealing from his job, but when I got home all the signs were there.";

break;

case 4:

dataString = serverTag + "JE " + user + " : I recently decided to sell my vacuum cleaner as all it was doing was gathering dust.";

break;

default:

System.out.println("Default");

dataString = defaultString;

collect.put(UUID+"true",0);

}

collect.put(UUID+"true",collect.get(UUID+"true")+1);

return dataString;

}

//Proverbs from http://www.u.arizona.edu/~rchaves/deepthought.html

/\*\*

\* Chooses Joke and proverb

\* @param UUID Unique String to store for random list

\* @return dataString String of Proverb

\*/

public String getProverb(String UUID){

String dataString ;

String unique = UUID+"false";

int selection = collect.get(unique);

if (selection == 5){

collect.put((UUID+"false"),0);

selection = 0;

}

List<Integer> messages = Arrays.asList(0, 1, 2, 3, 4);

//Get Seed Random List for random

messages = randomSeed(messages,(UUID+"false"));

//Collections.shuffle(messages, new Random(unique.hashCode()));

//System.out.println(messages);

String user = UUID.split(":")[1];

selection = messages.get(selection);

String serverTag = serverTagId(secondary);

String defaultString = serverTag + "PA " + user + " : Those who say it can't be done are usually interrupted by others doing it.";

switch (selection){

case 0:

dataString = defaultString;

break;

case 1:

dataString = serverTag + "PB " + user + " : A smile is an inexpensive way to change your looks.";

break;

case 2:

dataString = serverTag + "PC " + user + " : The pursuit of happiness is the chase of a life time.";

break;

case 3:

dataString = serverTag + "PD " + user + " : Every man dies; but not every man really lives.";

break;

case 4:

dataString = serverTag + "PE " + user + " : Better to understand little than to misunderstand a lot.";

break;

default:

dataString = defaultString;

collect.put(UUID+"false",0);

}

collect.put(UUID+"false",collect.get(UUID+"false")+1);

return dataString;

}

/\*\*

\* Checks if UUID(name) is found in List

\* @param name Unique String

\* @return Boolean if new UUID

\*/

public Boolean checkNew(String name){

if (collect.get(name) != null){

System.out.println("Old UUID :" + name);

return true;

}else{

collect.put(name,0);

return true;

}

}

/\*\*

\*Generate random list of jokes and proverb for each client.

\* @param list List that stores Random

\* @param UID

\* @return

\*/

public List<Integer> randomSeed(List<Integer> list,String UID){

//If value is found in list

if (uidCycle.get(UID) != null){

// System.out.println("Old Cycle UUID: " + UID);

//System.out.println("Old Cycle Key: " + uidCycle.get(UID) );

//System.out.println("Current Cycle Zero " + uidCycle.get(UID));

//If we reach end of joke/proverb cycle with 5 jokes/proverbs

if (uidCycle.get(UID) % 5 == 0){

//System.out.println("Current Cycle Zero " + uidCycle.get(UID));

//So benchmark that falls under

int real = uidCycle.get(UID)-1;

//System.out.println("Lower by one "+real);

//Generate seed by getting benchmark that substracting reminder from the value but make sure it falls

//under the previous benchmark

int seed = real - (real % 5);

//System.out.println("Seed is " + seed);

Collections.shuffle(list, new Random((UID+seed).hashCode()));

//Generate the benchmark by subtracting the reminder from the value

}else{

int seed = uidCycle.get(UID) - (uidCycle.get(UID) % 5);

//System.out.println("Current Cycle Less " + uidCycle.get(UID));

//System.out.println("Seed is " +seed);

Collections.shuffle(list, new Random((UID+seed).hashCode()));

}

uidCycle.put(UID, uidCycle.get(UID)+1);

//If value not found add to list ,recursive

}else{

//System.out.println("New Cycle");

uidCycle.put(UID,1);

return randomSeed(list , UID);

}

return list;

}

/\*\*

\* Tag to append to Joke/proverb output

\* @param check

\* @return String Tag

\*/

public String serverTagId(Boolean check){

if (check){

return "<s2> ";

}else{

return "<s1> ";

}

}

}