//Използване на библиотеката от класовете на Java за //входно – изходните потоци.

```
import java . oi . * ;
```

//Възможност за осъществяване на комуникации между //отделните компютри.

```
import java.net.*;
```

```
/**
```

*Създава се клас KnockKnockClient ;

```
**/
```

```
public class KnockKnockClient.{
```

```
/**
```

* Методът main е общо достъпен метод на класа *KnockKnockServer, който не връща резултат. *Изпълнението на програмата KnockKnockServer

*започва задължително с дефинирането на този *метод. Ако програмата не съдържа метода main тя *дава грешка.

*@param args[] Аргументите са от тип String.

**/

public static void main (String [ ] args) {

//Присвояваме null на обекта kkSocket от класа Socket.

Socket kkSocket = null ;

//Да отпечата изходния поток и да му се присвои //стойност null.

PrintStream os = null ;

//Входящи потоци; InputStream побайтово декориране с //Data InputStream.

DataInputStream is = null ;

//Ако имаме част от програмата , която е недостатъчно //надеждна в изпълнението си , то този фрагмент го //включва в оператора try във {…} скоби.

//Един и същи try може да има различни Exception .

try { kkSocket = new Soket (" Lab329s8,4444)

// Дефинира се променлива os от класа Stream (поток) . //getOutputstream -метод за побаитово четене на самият //поток , които е входящ.

os = new PrintStream (kkSocket . get OutpбutStream ( )) ;

//Дефинира се променлива os от класа Stream (поток). //getInputSream - метод за побайтово четене на самият поток , //който е изходящ.

is = new DataInputStream (kkSocket . getInputStream ( )) ; }

// Регистрира възникналото събитие и програмиста във скобите {…} го дефинира.

catch (UnknownHost Exception e) {

//Ако възникне системна грешка извежда на екран

//съобщението в скобите.

System . println (“Don’t know about host : Lab 329s8 “) ;

//Регистрира възникналото събитие и програмиста в

//скобите го дефинира {…}.

//IOException – входно-изходен .

catch (IOException e) {

//Ако възникне системна грешка извежда на екран

//съобщението в скобите (при възникнало вх.-изходно

//прекъсване) .

System.err.println (“Couldn’t get I/O for the con.to: Lab

329s8 ”) ; }

//Ако променливите kkSocket ;os ; is ; са различни от 0 да //изпълни следващият ред .

if (kkSoket ! = null && os ! = null && is ! = null) {

//Buffer е поле ; StringBuffer – името на стандартен клас,
//които е буфер на стрингове , с него определяме типа на
//променливата .Но за да бъде създадена в паметта на //компютарът променлива се използва конструктор на класа //StringBuffer(50), създаващ нов обект . new –оператор , //състоящ се в създаване на нов клас StringBuffer (50) . //Създава се буфер с дължина 50 .

try {StringBuffer buf = new StringBuffer (50) ;

//Дефинираме целочислената променлива *c .*

int c ; String from Server ;

//ReadLine - изчаква получената изходяща информация //до получаване на някаква входяща информация . Докато //садържанието на скобите е различно от null, да изпълни //следващият ред.

while ((from Server = is.readLine ( )) ! = null {

//System – клас на Java ; out – извежда потока към //стандартният изход на системата (екрана) .

//Определя се стринга , връща стойността му и след това я //разпечатва .

System . out . println (“Server : ” + from Server);

//Ако информацията е еквивалентна използваме от цикъла //while , ако не продължаваме .

if (from Server . equals (“Bye”)) break ;

//Докато не се въведе от клавиатурата нов ред да изпълни //слеващият ред.

while ((c = System . in read ( )) ; = ‘ n’) {

//Добавиаме в буфера поредният символ от

//клавиатурата .

buff. Append ((char) c) ; }

//System – клас на Java ; println – отпечатва ; out – извежда //потока към стандартният изход на системата (екрана) ;

//Показва какво е въвел потребителя от клавиатурата .

System .out . println (“Client : ” + buff) ;

//Отпечатва се изходящият поток .

os . println (buf . toString ( )) ; os fluch ( ) ;

//Даваме на буфера дължина null .

buf . setLength (0) ; }

//Затваря изходният поток .

os . close ( ) ;

//Затваря входящият поток .

is . close ( ) ;

//Затваря kkSocket .

kkSocket . close ( ) ; }

//Регистрира възникналото събитие и програмиста в скобите //го дефинира {… }.

catch (UnknownHost Exception e) {

//Ако възникне системна грешка извежда на екран //съобщението в скобите .

System.err.println (“Trying to conect to unknowshost; “+e);}

//e – обекта който се съсдава . Регистрира вх.- изходна //грешка.

catch (IOException e) {

//Ако възникне системна грешка извежда на екран //съобщението в скобите .

System . err . println (“ IOException ; “ + e) ; }}}}