实验报告

学院：计算机学院

专业：电子商务

姓名：李安琴

学号：0161123535

指导教师：苏向东

1. 实验目的

（1）按照作业要求完成每一步要求并提交。

（2）最后完成一个五子棋的AI。

1. 实验环境

Eclipse。

1. 实验使用语言

Java。

1. 实验步骤

## 第一步服务器是交战的战场，我们的AI大脑想要参战，先得找到去战场的路。

**public** **static** String getHTML(String urlToRead) **throws** Exception {

StringBuilder result = **new** StringBuilder();

URL url = **new** URL(urlToRead);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.setRequestMethod("GET");

BufferedReader rd = **new** BufferedReader(**new** InputStreamReader(conn.getInputStream()));

String line;

**while** ((line = rd.readLine()) != **null**) {

result.append(line);

}

rd.close();

**return** result.toString();

}

1. 请你告诉服务器你的名字是什么，你的学号是什么。名字使用字段name，学号使用字段student\_number。请把以上信息以GET方式交到以下地址：

System.***out***.println(*getHTML*("http://202.207.12.223:8000/step\_02?name=aaaqaaaaaa&student\_number=0161123535"));

1. 阅读以下材料，找出提示。
2. 实现模幂算法，通过服务器的检验。

**public** **static** BigInteger *a*=**new** BigInteger("328357472");

**public** **static** BigInteger *b*=**new** BigInteger("594991573");

**public** **static** BigInteger *c*=**new** BigInteger("187942120");

**public** **static** BigInteger *d*=**new** BigInteger("822832550");

**public** **static** BigInteger *e*=**new** BigInteger("2428");

**public** **static** BigInteger *f*=**new** BigInteger("828985668633");

**public** **static** BigInteger *g*=**new** BigInteger("3233");

**public** **static** BigInteger *h*=**new** BigInteger("904874541");

**public** **static** BigInteger *i*=**new** BigInteger("469972483897");

**public** **static** BigInteger *j*=**new** BigInteger("165124671");

System.***out***.println(*a*.modPow(**new** BigInteger("422491626"), **new** BigInteger("409553375")));

System.***out***.println(*b*.modPow(**new** BigInteger("883417601"), **new** BigInteger("314")));

System.***out***.println(*c*.modPow(**new** BigInteger("5454"), **new** BigInteger("9584")));

System.***out***.println(*d*.modPow(**new** BigInteger("973950441"), **new** BigInteger("637694487")));

System.***out***.println(*e*.modPow(**new** BigInteger("517402543754"), **new** BigInteger("241040558148")));

System.***out***.println(*f*.modPow(**new** BigInteger("124148072"), **new** BigInteger("66")));

System.***out***.println(*g*.modPow(**new** BigInteger("8135"), **new** BigInteger("147364890446")));

System.***out***.println(*h*.modPow(**new** BigInteger("357277884063"), **new** BigInteger("433690028")));

System.***out***.println(*i*.modPow(**new** BigInteger("294753086"), **new** BigInteger("256800708")));

System.***out***.println(*j*.modPow(**new** BigInteger("277291686889"), **new** BigInteger("740208595")));

实验结果：

64634529

299

6400

176532182

6846168988

45

79588877111

263643289

226407121

224825711

{"is\_success": true, "message": "please visit [http://202.207.12.223:8000/context/eb63fffd85c01a0a5d8f3cadea18cf56"}](http://202.207.12.223:8000/context/eb63fffd85c01a0a5d8f3cadea18cf56\"})

（5）把你的用户名作为user字段，加密后的密码作为password字段，提交到http://202.207.12.223:8000/step\_05

// 第5.1步

System.***out***.println(num.modPow(**new** BigInteger("65537"), **new** BigInteger("135261828916791946705313569652794581721330948863485438876915508683244111694485850733278569559191167660149469895899348939039437830613284874764820878002628686548956779897196112828969255650312573935871059275664474562666268163936821302832645284397530568872432109324825205567091066297960733513602409443790146687029")));

//第5.2步

String s="hello, world!";

BigInteger strnum2 =*str2num*( s);

System.***out***.println(strnum2);

//第5。3步

BigInteger str3=**new** BigInteger("8271117963530313756381553648673");

System.***out***.println(str3.modPow(**new** BigInteger("65537"), **new** BigInteger("135261828916791946705313569652794581721330948863485438876915508683244111694485850733278569559191167660149469895899348939039437830613284874764820878002628686548956779897196112828969255650312573935871059275664474562666268163936821302832645284397530568872432109324825205567091066297960733513602409443790146687029")));

//第5.4步

BigInteger str4=**new** BigInteger("41717036756922086167767786786712423269418006844571540642506559265291791915434146228186913317848309015094527019477424371059092802944619669306914382602706827360526972065492061812094685980946805703427198780190727908992795654922187272049190698839062058797703366041101170679313306423879973490422740979302379859262");

System.***out***.println(str4.toString(16));

//第5.5步

String str5="760817";

BigInteger str5\_=*str2num*(str5);

BigInteger str5\_5=(str5\_.modPow(**new** BigInteger("65537"), **new** BigInteger("135261828916791946705313569652794581721330948863485438876915508683244111694485850733278569559191167660149469895899348939039437830613284874764820878002628686548956779897196112828969255650312573935871059275664474562666268163936821302832645284397530568872432109324825205567091066297960733513602409443790146687029")));

System.***out***.println(str5\_5.toString(16));

**public** **static** BigInteger str2num(String s)//转换为256进制

{

s=**new** StringBuffer(s).reverse().toString();

BigInteger total=**new** BigInteger("0");

**int** l=s.length();

**for**(**int** i=0;i<l;i++){

**char** c=s.charAt(i);

BigInteger m=**new** BigInteger("256");

m=m.pow(i);

BigInteger n=**new** BigInteger(String.*valueOf*((**int**)(c)));

total=total.add(m.multiply(n));

}

**return** total;

}

（6）实现从坐标表示到图形表示的算法，通过服务器的检验。

String str6="ggffhggfhffgiefhfeheidehdidhghegcidfgiefcfeeeied";

*GetMap*(str6);

**public** **static** StringBuilder GetMap(String s) //得到每一步棋的地图6

{

StringBuilder sb = **new** StringBuilder();

StringBuilder sb1 = **new** StringBuilder();

String s1 = **new** String();

**int** temp = 1;

**for** (**int** j = 0; j < 15; j++) { //初始化棋盘

**for** (**int** k = 0; k < 15; k++) {

sb1.append(".");

}

}

s1 = sb1.toString();

**for** (**int** i = 0; i < s.length(); i += 2) {

**char** a = s.charAt(i);

**char** b = s.charAt(i + 1);

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

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

**int** index = (a - 'a') \* 15 + b - 'a';

**if** (temp == 1) {

sb1.replace(index, index + 1, "x");

temp = 2;

} **else** **if** (temp == 2) {

sb1.replace(index, index + 1, "o");

temp = 1;

}

sb.append(",");

sb.append(sb1.toString());

s1 = sb1.toString();

sb1 = **new** StringBuilder();

sb1.append(s1);

}

//sb.deleteCharAt(sb.length()-1);

sb.deleteCharAt(0); //删掉第一个

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

**return** sb;

}

................................................................................................x................................................................................................................................,................................................................................o...............x................................................................................................................................,................................................................................o...............x..............x.................................................................................................................,................................................................................o..............ox..............x.................................................................................................................,................................................................................o..............ox.............xx.................................................................................................................,................................................................................oo.............ox.............xx.................................................................................................................,................................................................................oo.............ox.............xx............x....................................................................................................,................................................................................ooo............ox.............xx............x....................................................................................................,...............................................................................xooo............ox.............xx............x....................................................................................................,...............................................................................xooo............ox............oxx............x....................................................................................................,...............................................................................xooo............ox............oxx...........xx....................................................................................................,...................................................................o...........xooo............ox............oxx...........xx....................................................................................................,.....................................................x.............o...........xooo............ox............oxx...........xx....................................................................................................,....................................................ox.............o...........xooo............ox............oxx...........xx....................................................................................................,....................................................ox.............o...........xooo............oxx...........oxx...........xx....................................................................................................,....................................................ox............oo...........xooo............oxx...........oxx...........xx....................................................................................................,......................................x.............ox............oo...........xooo............oxx...........oxx...........xx....................................................................................................,......................................x...........o.ox............oo...........xooo............oxx...........oxx...........xx....................................................................................................,......................................x...........o.ox............oo...........xooo............oxxx..........oxx...........xx....................................................................................................,......................................x...........o.ox...........ooo...........xooo............oxxx..........oxx...........xx....................................................................................................,...................................x..x...........o.ox...........ooo...........xooo............oxxx..........oxx...........xx....................................................................................................,...................................x..x...........o.ox..........oooo...........xooo............oxxx..........oxx...........xx....................................................................................................,...................................x..x...........o.ox..........oooox..........xooo............oxxx..........oxx...........xx....................................................................................................,...................................x..x...........o.ox.........ooooox..........xooo............oxxx..........oxx...........xx....................................................................................................

{"is\_success": true, "message": "please visit http://202.207.12.223:8000/context/c03152f5db8918b9905d449686685f77"}

（7）实现棋型算法，通过服务器的检验。

String board="ghhggggfgigjhhfhiijjfghiihigjhkhhjgkjgffhfkgjijfiejdjeheidkfkikekdlfmgmfifnf";

String coord[]=**new** String[] {"nk", "ok", "bn", "ef", "lk", "im", "gb", "fd", "ha"};

**char** newboard[][]=*get7*(board);

*get77*(newboard,coord);

**public** **static** **char**[][] get7(String board) {

StringBuilder sc=**new** StringBuilder();

**int** temp=1;

**for**(**int** i=0;i<225;i++)

{

sc.append(".");

}

**for**(**int** i=0;i<board.length();i+=2) {

**char** a=board.charAt(i);

**char** b=board.charAt(i+1);

**int** index = (a - 'a') \* 15 + b - 'a';

**if** (temp == 1) {

sc.replace(index, index + 1, "x");

temp = 2;

} **else** **if** (temp == 2) {

sc.replace(index, index + 1, "o");

temp = 1;

}

}

//System.out.println(sc);

String sc\_1;

sc\_1=sc.toString();

**char** changea[]=sc\_1.toCharArray();//利用toCharArray方法转换

**char** changeb[][]=**new** **char**[15][15];

**int** c=0;

**for**(**int** i=0;i<15;i++)

{

**for**(**int** j=0;j<15;j++)

{

changeb[i][j]=changea[c];

c++;

//System.out.println(changeb[i][j]);

}

}

**return** changeb;

}

**public** **static** **void** get77(**char** nboard[][],String coord[]){

**int** m ;

**int** x = 0, y = 0;

**for** (m = 0; m < coord.length; m++)

{

String s = coord[m];

**int** yy = 0;

**while** (yy < s.length() - 1)

{

**int** s2,s4;

x = s.charAt(yy) - 'a';//转换数字

y = s.charAt(yy + 1) - 'a';

**for** (s2 = x, s4 = y - 4; s4 <=y + 4; s4++)

{

**if** ((s2 >= 0 && s2 < 15) && (s4 >= 0 && s4 < 15))

{

System.***out***.print(nboard[s2][s4]);

}

}

System.***out***.print(",");

yy+=2;

**for** (s2 = x - 4, s4 = y - 4; s4 <= y + 4 && s2 <= x + 4; s4++, s2++)

{

**if** ((s2 >= 0 && s2 < 15) && (s4 >= 0 && s4 < 15))

{

System.***out***.print(nboard[s2][s4]);

}

}

System.***out***.print(",");

**for** (s2 = x - 4, s4 = y; s2 <= x + 4; s2++)

{

**if** ((s2 >= 0 && s2 < 15) && (s4 >= 0 && s4 < 15))

{

System.***out***.print(nboard[s2][s4]);

}

}

System.***out***.print(",");

**for** (s2 = x + 4, s4 = y - 4; s4 <= y + 4 && s2>=x - 4; s4++, s2--)

{

**if** ((s2 >= 0 && s2 < 15) && (s4 >= 0 && s4 < 15))

{

System.***out***.print(nboard[s2][s4]);

}

}

**if**(m<coord.length-1)

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

}

}

}

.........,xo....,......,......,

.........,o....,.....,.....,

......,...,......,......,

.........,.....xxo.,.....ooxx,.........,

.........,oxx.....,........,........,

x......,..o....,.........,.......,

.....o,...xxo,.........,......,

.....oxo,.....xox,.......xo,........,

....o,...x.,.........,.....

{"is\_success": true, "message": "please visit [http://202.207.12.223:8000/context/701672826a45d8d2d998b3a3f66166bf"}](http://202.207.12.223:8000/context/701672826a45d8d2d998b3a3f66166bf\"})

（8）实现按照评分规则选择落子位置的算法，通过服务器的检验。

**public** **static** **int** score(String s) //获得一个点一条线的权重

{

**int** sum = 0;

String str1\_1 = **new** String();

str1\_1 = "CMMMM";

String str1\_2 = **new** String();

str1\_2 = "MCMMM";

String str1\_3 = **new** String();

str1\_3 = "MMCMM";

String str1\_4 = **new** String();

str1\_4 = "MMMCM";

String str1\_5 = **new** String();

str1\_5 = "MMMMC";

**int** scr1 = 10000;

**if** (s.indexOf(str1\_1) > -1) {

sum = sum + scr1;

}

**if** (s.indexOf(str1\_2) > -1) {

sum = sum + scr1;

}

**if** (s.indexOf(str1\_3) > -1) {

sum = sum + scr1;

}

**if** (s.indexOf(str1\_4) > -1) {

sum = sum + scr1;

}

**if** (s.indexOf(str1\_5) > -1) {

sum = sum + scr1;

}

String str2\_1 = **new** String();

str2\_1 = "OOOOC";

String str2\_2 = **new** String();

// str2\_2 = "COOOO";

// String str2\_3 = new String();

// str2\_3 = "OCOOO";

// String str2\_4 = new String();

// str2\_4 = "OOCOO";

// String str2\_5 = new String();

// str2\_5 = "OOOCO";

// int scr2 = 8000;

**int** scr2 = 8000;

**if** (s.indexOf(str2\_1) > -1) {

sum = sum + scr2;

}

**if** (s.indexOf(str2\_2) > -1) {

sum = sum + scr2;

}

// if (s.indexOf(str2\_3) > -1) {

// sum = sum + scr2;

// }

// if (s.indexOf(str2\_4) > -1) {

// sum = sum + scr2;

// }

// if (s.indexOf(str2\_5) > -1) {

// sum = sum + scr2;

// }

String str3\_1 = **new** String();

str3\_1 = ".CMMM.";

String str3\_2 = **new** String();

str3\_2 = ".MCMM.";

String str3\_3 = **new** String();

str3\_3 = ".MMCM.";

String str3\_4 = **new** String();

str3\_4 = ".MMMC.";

**int** scr3 = 5000;

**if** (s.indexOf(str3\_1) > -1) {

sum = sum + scr3;

}

**if** (s.indexOf(str3\_2) > -1) {

sum = sum + scr3;

}

**if** (s.indexOf(str3\_3) > -1) {

sum = sum + scr3;

}

**if** (s.indexOf(str3\_4) > -1) {

sum = sum + scr3;

}

String str4\_1 = **new** String();

str4\_1 = "COOO.";

String str4\_2 = **new** String();

str4\_2 = ".OOOC";

String str4\_3 = **new** String();

str4\_3 = ".OOCO.";

String str4\_4 = **new** String();

str4\_4 = ".OCOO.";

// int scr4 = 3500;

**int** scr4 = 2500;

**if** (s.indexOf(str4\_1) > -1) {

sum = sum + scr4;

}

**if** (s.indexOf(str4\_2) > -1) {

sum = sum + scr4;

}

**if** (s.indexOf(str4\_3) > -1) {

sum = sum + scr4;

}

**if** (s.indexOf(str4\_4) > -1) {

sum = sum + scr4;

}

String str5\_1 = **new** String();

str5\_1 = "OCMMM.";

String str5\_2 = **new** String();

str5\_2 = "OMCMM.";

String str5\_3 = **new** String();

str5\_3 = "OMMCM.";

String str5\_4 = **new** String();

str5\_4 = "OMMMC.";

String str5\_5 = **new** String();

str5\_5 = ".CMMMO";

String str5\_6 = **new** String();

str5\_6 = ".MCMMO";

String str5\_7 = **new** String();

str5\_7 = ".MMCMO";

String str5\_8 = **new** String();

str5\_8 = ".MMMCO";

**int** scr5 = 2000;

**if** (s.indexOf(str5\_1) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_2) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_3) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_4) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_5) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_6) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_7) > -1) {

sum = sum + scr5;

}

**if** (s.indexOf(str5\_8) > -1) {

sum = sum + scr5;

}

String str6\_1 = **new** String();

str6\_1 = ".MMC.";

String str6\_2 = **new** String();

str6\_2 = ".MCM.";

String str6\_3 = **new** String();

str6\_3 = ".CMM.";

**int** scr6 = 400;

**if** (s.indexOf(str6\_1) > -1) {

sum = sum + scr6;

}

**if** (s.indexOf(str6\_2) > -1) {

sum = sum + scr6;

}

**if** (s.indexOf(str6\_3) > -1) {

sum = sum + scr6;

}

String str7\_1 = **new** String();

str7\_1 = ".OOC";

String str7\_2 = **new** String();

str7\_2 = "COO.";

String str7\_3 = **new** String();

str7\_3 = "MOOOC";

String str7\_4 = **new** String();

str7\_4 = "COOOM";

// String str7\_5 = new String();

// str7\_5 = ".OCO";

// String str7\_6 = new String();

// str7\_6 = "OCO.";

**int** scr7 = 400;

**if** (s.indexOf(str7\_1) > -1) {

sum = sum + scr7;

}

**if** (s.indexOf(str7\_2) > -1) {

sum = sum + scr7;

}

**if** (s.indexOf(str7\_3) > -1) {

sum = sum + scr7;

}

**if** (s.indexOf(str7\_4) > -1) {

sum = sum + scr7;

}

// if (s.indexOf(str7\_5) > -1) {

// sum = sum + scr7;

// }

// if (s.indexOf(str7\_6) > -1) {

// sum = sum + scr7;

// }

String str8\_1 = **new** String();

str8\_1 = ".MMCO";

String str8\_2 = **new** String();

str8\_2 = ".MCMO";

String str8\_3 = **new** String();

str8\_3 = ".CMMO";

String str8\_4 = **new** String();

str8\_4 = "OMMC.";

String str8\_5 = **new** String();

str8\_5 = "OMCM.";

String str8\_6 = **new** String();

str8\_6 = "OCMM.";

String str8\_7 = **new** String();

str8\_7 = "MOOC";

String str8\_8 = **new** String();

str8\_8 = "COOM";

**int** scr8 = 200;

**if** (s.indexOf(str8\_1) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_2) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_3) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_4) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_5) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_6) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_7) > -1) {

sum = sum + scr8;

}

**if** (s.indexOf(str8\_8) > -1) {

sum = sum + scr8;

}

String str9\_1 = **new** String();

str9\_1 = ".MC.";

String str9\_2 = **new** String();

str9\_2 = ".CM.";

// int scr9 = 100;

**int** scr9 = 50;

**if** (s.indexOf(str9\_1) > -1) {

sum = sum + scr9;

}

**if** (s.indexOf(str9\_2) > -1) {

sum = sum + scr9;

}

//

// String str10\_1 = new String();

// str10\_1 = ".OC.";

// String str10\_2 = new String();

// str10\_2 = ".CO.";

// int scr10 = 50;

// if (s.indexOf(str10\_1) > -1) {

// sum = sum + scr10;

// }

// if (s.indexOf(str10\_2) > -1) {

// sum = sum + scr10;

// }

**int** scr\_other = 20;

**if** (sum == 0) {

sum = sum + scr\_other;

}//其他棋型

**return** sum;

}

ki

mj

ih

le

hg

ia

eh

gi

ci

hi

ke

kh

gl

gm

hi

kh

hj

{"is\_success": true, "message": "please visit [http://202.207.12.223:8000/context/2e329756827e42330f7897cc9499588e"}](http://202.207.12.223:8000/context/2e329756827e42330f7897cc9499588e\"})

（9）实现对战。

1. 实验总结

在此过程中，我利用这个机会学习了Java，虽然还不是全面了解，但已可以实现基本编程，由于Java与C++有很多相似的地方，学习过程较为顺利。其中实践运用过程遇到了很多问题，通过与同学的交流、实践，都已得到解决，算法也经过了基本优化。