

## 前回の解答

step2 の最後に下のような注意書きがありました。

### 注意

このプログラムは、スタート位置が右の壁際の場合、上の壁まで移動したあと壁にぶつかってしまいます。

これを回避するにはプログラム中で判断を追加しなければなりません。どこにどのような判断を追加したらよいか考えてみてください。

これは、上の壁まで移動した後、何も判断しないで右に移動してしまうため起こります。これを回避するには次のようなプログラムを追加します（赤字が追加したところです）。こうすることで、上の壁まで来たとき右に移動できるか判断をするので右の壁にぶつからないようになります。

```

18      /****** mode で分岐する *****/
19      switch (mode) {
20          case 1: /* 壁にぶつかるまで上に移動する */
21              if (value[2] == 2) { /* 上に壁があったら */
22                  if (value[6] == 2) { /* 右に壁があるか判断する */
23                      value = target.walkDown();
24                      mode = 3;
25                  }
26                  else {
27                      value = target.walkRight(); /* 右に移動し */
28                      mode = 2; /* mode を 2 に変更する */
29                  }
30              }
31          else {
32              value = target.walkUp(); /* そうでなければ、上に移動する */
33          }
34          break;
35          case 2: /* 壁にぶつかるまで右に移動する */
36              .
37              .
38              .

```

### サンプルプログラム (Sample2008\_03. java)

- ・まず、上と右をサーチし、自分の位置を把握します
- ・次に、現在位置から一番近い四隅（右上、右下、左上、左下）を割り出し、そこに移動します
- ・その後、“ Sample2008\_02. java” と同じ動き（壁の内側を時計回りに移動）をします
- ・新たに step という変数を作り、現在実行している mode の中で順序（step）をわけて管理します
- ・プログラム名は“ Sample2008\_03. java” とします
- ・赤字が追加・変更したところです

```

1 public class Sample2008_03{
2     public static void main(String[ ] args){
3
4         int[ ] value;
5         value = new int[10];
6
7         int mode = 5,
8             step = 1,
9             posY = 1,
10            posX = 1;
11
12        /***** 競技サーバに接続する *****/
13        edu.procon.PreConnect2008 target;
14        target = new edu.procon.PreConnect2008( );
15
16        while(true){
17            /***** まず、getReady でサーバから戻り値を得る *****/
18            value = target.getReady( );
19            if(value[0] == 0) break;
20
21            /***** mode で分岐する *****/
22            switch (mode){
23                case 1: /* 壁にぶつかるまで上に移動する */
24                    if(value[2] == 2){ /* 上に壁があったら */
25                        if(value[6] == 2){ /* 右に壁があるか判断する */
26                            value = target.walkDown( );
27                            mode = 3;
28                        }
29                        else{
30                            value = target.walkRight( ); /* 右に移動し */
31                            mode = 2; /* mode を 2 に変更する */
32                        }
33                    }
34                    else{
35                        value = target.walkUp( ); /* そうでなければ、上に移動する */
36                    }
37                    break;
38                case 2: /* 壁にぶつかるまで右に移動する */
39                    if(value[6] == 2){

```

```

40         value = target.walkDown( );
41         mode = 3;
42     }
43     else{
44         value = target.walkRight( );
45     }
46     break;
47 case 3: /* 壁にぶつかるまで下に移動する */
48     if(value[8] == 2) {
49         value = target.walkLeft( );
50         mode = 4;
51     }
52     else{
53         value = target.walkDown( );
54     }
55     break;
56 case 4: /* 壁にぶつかるまで左に移動する */
57     if(value[4] == 2) {
58         value = target.walkUp( );
59         mode = 1;
60     }
61     else{
62         value = target.walkLeft( );
63     }
64     break;
65 case 5: /* 上と右をサーチし、一番近い四隅へ移動する */
66     switch (step){
67         case 1: /* step1 上をサーチする */
68             value = target.searchUp( );
69             for(int i=1; i<10; i++){
70                 if(value[i] == 2) {
71                     posY = i; /* 現在位置の Y 座標が決定 */
72                     break;
73                 }
74             }
75             step = 2;
76             break;
77         case 2: /* step2 右をサーチする */
78             value = target.searchRight( );
79             for(int i=1; i<10; i++){
80                 if(value[i] == 2) {
80                     posX = 10 - i; /* 現在位置の X 座標が決定 */
81                     break;
82                 }
83             }
84             step = 3;
85             break;
86         case 3: /* step3 一番近い四隅を見つける */
87             if(posY<=5 && posX<=5) {
88                 if(value[2] != 2) { /* 上に動くことができるか判断する */
89                     value = target.walkUp( );

```

```

90         mode = 6; /* 左上に移動するモードへ */
91         step = 1;
92     }
93     else if(value[4] != 2) { /* 左に動くことができるか判断する */
94         value = target.walkLeft( );
95         mode = 6; /* 左上に移動するモードへ */
96         step = 2;
97     }
98     else { /* すでに左上にいるので、右に移動する */
99         value = target.walkRight( );
100         mode = 2; /* 右に移動するモードへ */
101     }
102 }
103 else if(posY<=5 && posX>=6) {
104     if(value[2] != 2) { /* 上に動くことができるか判断する */
105         value = target.walkUp( );
106         mode = 7; /* 右上に移動するモードへ */
107         step = 1;
108     }
109     else if(value[6] != 2) { /* 右に動くことができるか判断する */
110         value = target.walkRight( );
111         mode = 7; /* 右上に移動するモードへ */
112         step = 2;
113     }
114     else { /* すでに右上にいるので、下に移動する */
115         value = target.walkDown( );
116         mode = 3; /* 下に移動するモードへ */
117     }
118 }
119 else if(posY>=6 && posX<=5) {
120     if(value[8] != 2) { /* 下に動くことができるか判断する */
121         value = target.walkDown( );
122         mode = 8; /* 左下に移動するモードへ */
123         step = 1;
124     }
125     else if(value[4] != 2) { /* 左に動くことができるか判断する */
126         value = target.walkLeft( );
127         mode = 8; /* 左下に移動するモードへ */
128         step = 2;
129     }
130     else { /* すでに左下にいるので、上に移動する */
131         value = target.walkUp( );
132         mode = 1; /* 上に移動するモードへ */
133     }
134 }
135 else {
136     if(value[8] != 2) { /* 下に動くことができるか判断する */
137         value = target.walkDown( );
138         mode = 9; /* 右下に移動するモードへ */
139         step = 1;
140     }

```

```

141         else if(value[6] != 2) { /* 右に動くことができるか判断する */
142             value = target.walkLeft( );
143             mode = 9; /* 右下に移動するモードへ */
144             step = 2;
145         }
146         else { /* すでに右下にいるので、左に移動する */
147             value = target.walkLeft( );
148             mode = 4; /* 左に移動するモードへ */
149         }
150     }
151     break;
152 }
153 break;
154 case 6: /* 左上に移動するモード */
155     switch (step) {
156         case 1: /* step1 壁にぶつかるまで上に移動する */
157             if(value[2] == 2) {
158                 if(value[4] == 2) { /* 左に壁があるか判断する */
159                     value = target.walkRight( );
160                     mode = 2; /* 壁にぶつかるまで右に移動するモードへ */
161                     step = 1;
162                 }
163                 else {
164                     value = target.walkLeft( );
165                     step = 2;
166                 }
167             }
168             else {
169                 value = target.walkUp( );
170             }
171             break;
172         case 2: /* step2 壁にぶつかるまで左に移動する */
173             if(value[4] == 2) {
174                 value = target.walkRight( );
175                 mode = 2; /* 壁にぶつかるまで右に移動するモードへ */
176                 step = 1;
177             }
178             else {
179                 value = target.walkLeft( );
180             }
181             break;
182     }
183     break;
184 case 7: /* 右上に移動するモード */
185     switch (step) {
186         case 1: /* step1 壁にぶつかるまで上に移動する */
187             if(value[2] == 2) {
188                 if(value[6] == 2) {
189                     value = target.walkDown( );
190                     mode = 3; /* 壁にぶつかるまで下に移動するモードへ */
191                     step = 1;

```

```

192         }
193         else{
194             value = target.walkRight( );
195             step = 2;
196         }
197     }
198     else{
199         value = target.walkUp( );
200     }
201     break;
202 case 2: /* step2 壁にぶつかるまで右に移動する */
203     if(value[6] == 2) {
204         value = target.walkDown( );
205         mode = 3; /* 壁にぶつかるまで下に移動するモードへ */
206         step = 1;
207     }
208     else{
209         value = target.walkRight( );
210     }
211     }
212     break;
213 }
214 break;
215 case 8: /* 左下に移動するモード */
216 switch (step) {
217     case 1: /* step1 壁にぶつかるまで下に移動する */
218         if(value[8] == 2) {
219             if(value[4] == 2) {
220                 value = target.walkUp( );
221                 mode = 1; /* 壁にぶつかるまで上に移動するモードへ */
222                 step = 1;
223             }
224             else{
225                 value = target.walkLeft( );
226                 step = 2;
227             }
228         }
229         else{
230             value = target.walkDown( );
231         }
232         break;
233     case 2: /* step2 壁にぶつかるまで左に移動する */
234         if(value[4] == 2) {
235             value = target.walkUp( );
236             mode = 1; /* 壁にぶつかるまで上に移動するモードへ */
237             step = 1;
238         }
239         else{
240             value = target.walkLeft( );
241         }
242         break;
243     }

```

```

244         break;
245     case 9: /* 右下に移動するモード */
246         switch (step) {
247             case 1: /* step1 壁にぶつかるまで下に移動する */
248                 if(value[8] == 2) {
249                     if(value[6] == 2) {
250                         value = target.walkLeft( );
251                         mode = 4; /* 壁にぶつかるまで左に移動するモードへ */
252                         step = 1;
253                     }
254                     else{
255                         value = target.walkRight( );
256                         step = 2;
257                     }
258                 }
259                 else{
260                     value = target.walkDown( );
261                 }
262                 break;
263             case 2: /* step2 壁にぶつかるまで右に移動する */
264                 if(value[6] == 2) {
265                     value = target.walkLeft( );
266                     mode = 4; /* 壁にぶつかるまで左に移動するモードへ */
267                     step = 1;
268                 }
269                 else{
270                     value = target.walkRight( );
271                 }
272                 break;
273             }
274         break;
275     } /* switch(mode) */
276
277     /****** 制御情報が 0 だったら終了する *****/
278     if(value[0] == 0) break;
279
280 } /* while(true) */
281
282 /****** 競技サーバから切断する *****/
283 target.exit( );
284 }
285 }
286

```

※ 長いプログラムですが、がんばりましょう！

## プログラム説明

### ①変数の宣言と初期値

```

7      int mode = 5,
8          step = 1,
9          posY = 1,
10         posX = 1;

```

**mode** は上と右をサーチし、一番近い四隅へ移動するモードにするので初期値は **5** になります。**step** は 1 から始まるので **1** とします。**posY** は自分の Y 座標の現在位置、**posX** は自分の X 座標の現在位置なので初期値はそれぞれ **1** とします。

### ②追加したモードとステップ

今回新たにモード **5** から **9** を追加しました。ステップはモード内の順序を表しています。例えば、モード **5** の場合、ステップ **1** は上をサーチする、ステップ **2** は右をサーチする、ステップ **3** は一番近い四隅を見つけるというようになります。

#### ・モードとステップの種類と動作

mode	動作	
1	壁にぶつかるまで上に移動する	
2	壁にぶつかるまで右に移動する	
3	壁にぶつかるまで下に移動する	
4	壁にぶつかるまで左に移動する	
5	上と右をサーチし、一番近い四隅へ移動する	
	step = 1	上をサーチする
	step = 2	右をサーチする
	step = 3	一番近い四隅を見つける
6	左上に移動するモード	
	step = 1	壁にぶつかるまで上に移動する
	step = 2	壁にぶつかるまで左に移動する
7	右上に移動するモード	
	step = 1	壁にぶつかるまで上に移動する
	step = 2	壁にぶつかるまで右に移動する
8	左下に移動するモード	
	step = 1	壁にぶつかるまで下に移動する
	step = 2	壁にぶつかるまで左に移動する
9	右下に移動するモード	
	step = 1	壁にぶつかるまで下に移動する
	step = 2	壁にぶつかるまで右に移動する



## ③モード 5

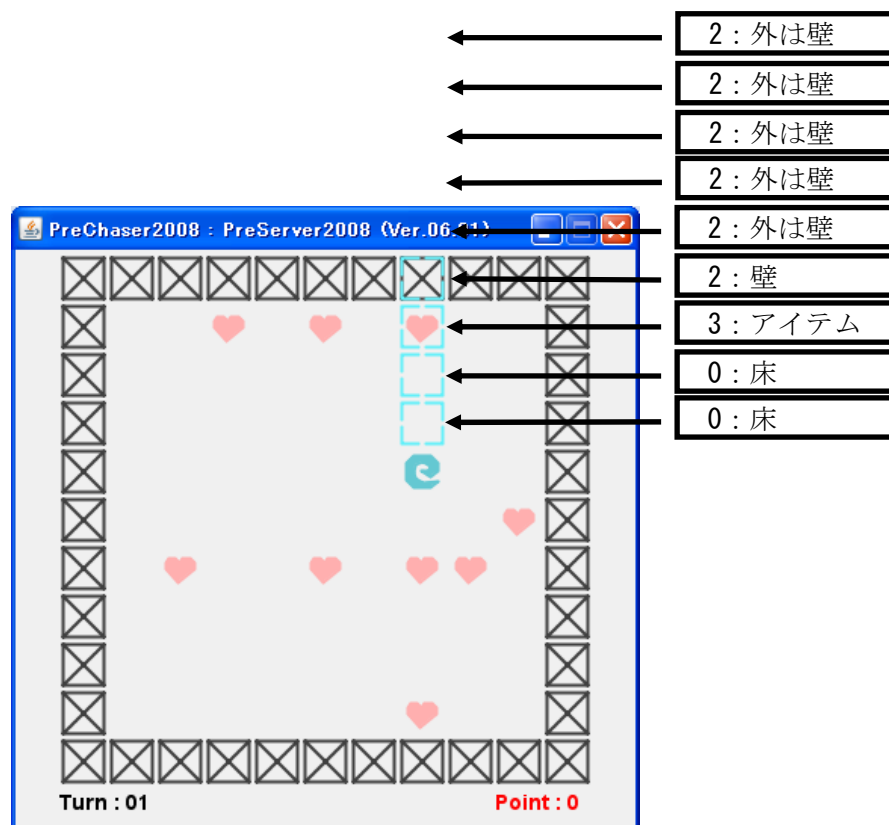
モード 5 はステップ 1 から 3 に分かれるので mode の switch 文の中に step の switch 文が入ります。ステップ 1 は上をサーチして Y 座標の位置を調べるので、まず serchUp メソッドを送信します。PreChaser は 9×9 マスなので serchUp メソッドを送信すれば必ず壁を検知します。壁の値は 2 なので、69 行目の for 文で Y 座標の位置を求めることができます。

```

65      case 5: /* 上と右をサーチし、一番近い四隅へ移動する */
66          switch (step){
67              case 1: /* step1 上をサーチする */
68                  value = target.searchUp();
69                  for(int i=1; i<10; i++){
70                      if(value[i] == 2){
71                          posY = i; /* 現在位置の Y 座標が決定 */
72                          break;
73                      }
74                  }
75                  step = 2;
76                  break;

```

例えば、下の画面の位置で serchUp メソッドを送信した場合、サーバからの戻り値は 1003222222 となります。先頭の 1 は制御情報で、それ以降は自分のキャラクタの上方向に周囲情報になります。壁の外側はすべて壁になっています。この場合は posY は 4 になります。



ステップ 2 は右をサーチして X 座標の位置を調べるので、まず `serchRight` メソッドを送信します。同様に 79 行目の `for` 文で X 座標の位置を求めることができます。

```

77         case 2: /* step2 右をサーチする */
78             value = target.searchRight();
79             for(int i=1; i<10; i++){
80                 if(value[i] == 2){
80                     posX = 10 - i; /* 現在位置の X 座標が決定 */
81                     break;
82                 }
83             }
84             step = 3;
85             break;

```

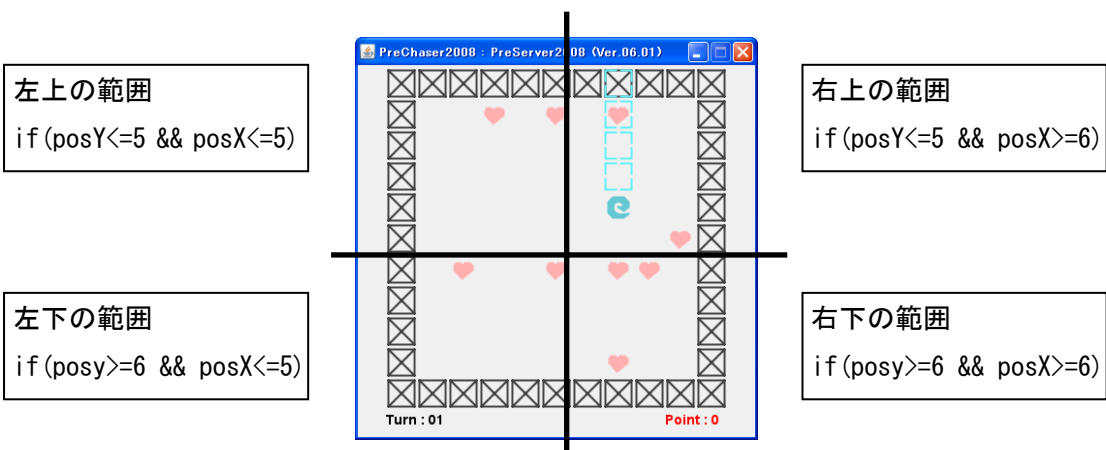
例えば、上の画面の位置で `serchRight` メソッドを送信した場合、サーバからの戻り値は 1002222222 となるので `posX` は  $10-3$  で 7 になります。

ステップ 3 は `posY` と `posX` を使って現在位置から一番近い四隅を割り出します。87 行目の `if` 文は `posY<=5` かつ `posX<=5` なので、現在位置がフィールドの左上の範囲にいるかを判断しています。同様に右上、左下、右下を判断します。

```

86         case 3: /* step3 一番近い四隅を見つける */
87             if(posY<=5 && posX<=5){
88                 if(value[2] != 2){ /* 上に動くことができるか判断する */
89                     value = target.walkUp();
90                     mode = 6; /* 左上に移動するモードへ */
91                     step = 1;
92                 }
93             } else if(value[4] != 2){ /* 左に動くことができるか判断する */
94                 value = target.walkLeft();
95                 mode = 6; /* 左上に移動するモードへ */
96                 step = 2;
97             }
98             else{ /* すでに左上にいるので、右に移動する */
99                 value = target.walkRight();
100                 mode = 2; /* 右に移動するモードへ */
101             }
102         }

```



## ④モード 6

モード 6 は左上に移動するモードなので、ステップ 1 で上に移動し、ステップ 2 で左に移動します。

```

154         case 6: /* 左上に移動するモード */
155             switch (step) {
156                 case 1: /* step1 壁にぶつかるまで上に移動する */
157                     if(value[2] == 2) {
158                         if(value[4] == 2) { /* 左に壁があるか判断する */
159                             value = target.walkRight();
160                             mode = 2; /* 壁にぶつかるまで右に移動するモードへ */
161                             step = 1;
162                         }
163                         else{
164                             value = target.walkLeft();
165                             step = 2;
166                         }
167                     }
168                 else{
169                     value = target.walkUp();
170                 }
171                 break;
172                 case 2: /* step2 壁にぶつかるまで左に移動する */
173                     if(value[4] == 2) {
174                         value = target.walkRight();
175                         mode = 2; /* 壁にぶつかるまで右に移動するモードへ */
176                         step = 1;
177                     }
178                     else{
179                         value = target.walkLeft();
180                     }
181                     break;
182             }
183             break;

```

同様に、モード 7 から 9 をつくります。