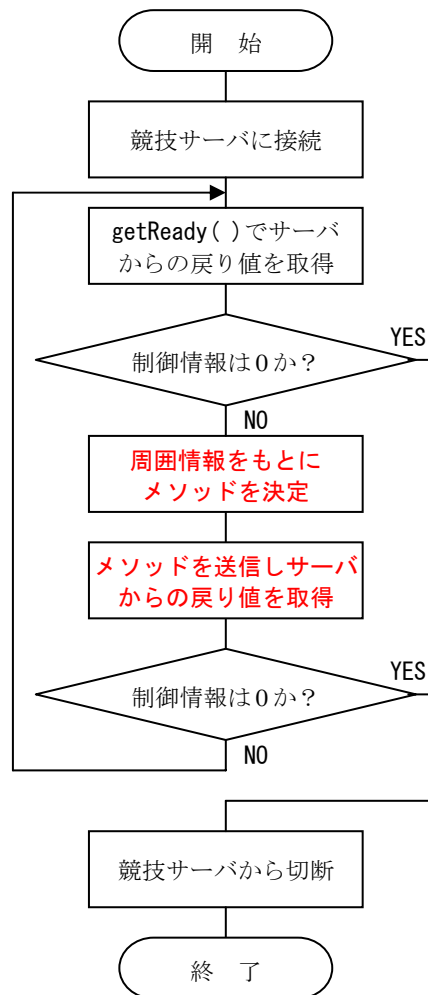


## プログラムの流れ

競技参加者の皆さんがつくるクライアントプログラムと競技サーバのやりとりは、`getReady` メソッドを送信して周囲情報を取得した後、動作メソッド(walk系、look系、search系)を選んで送信するという動きを繰り返します。

ほとんどの場合、クライアントプログラムは下のフローチャートのようになります。

競技参加者の皆さんは、サーバから得た周囲情報をもとにメソッドを選んで送信する部分を考えてください。



PreCHaser2008 は対戦型ではないので、フィールドに置かれたアイテムを制限ターン内で取ることを目標に考えます。

### サンプルプログラム (Sample2008\_02. java)

- ・ まず、壁にぶつかるまで上に移動します
- ・ 次に右に移動し、壁の内側を時計回りに移動します
- ・ ターン制限 (81 ターン) まで移動し、ゲームオーバーになります
- ・ mode という変数を作り、動作の種類を記憶させます
- ・ プログラム名は” Sample2008\_02. java” とします

```

1 public class Sample2008_02{
2     public static void main(String[ ] args){
3
4         int[ ] value;
5         value = new int[10];
6
7         int mode = 1;
8
9         /***** 競技サーバに接続する *****/
10        edu.procon.PreConnect2008 target;
11        target = new edu.procon.PreConnect2008( );
12
13        while(true){
14            /***** まず、getReady でサーバから戻り値を得る *****/
15            value = target.getReady( );
16            if(value[0] == 0) break;
17
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                        value = target.walkRight( ); /* 右に移動し */
23                        mode = 2; /* mode を 2 に変更する */
24                    }
25                    else{
26                        value = target.walkUp( ); /* そうでなければ、上に移動する */
27                    }
28                    break;
29                case 2: /* 壁にぶつかるまで右に移動する */
30                    if(value[6] == 2){
31                        value = target.walkDown( );
32                        mode = 3;
33                    }
34                    else{
35                        value = target.walkRight( );
36                    }
37                    break;
38                case 3: /* 壁にぶつかるまで下に移動する */
39                    if(value[8] == 2){
40                        value = target.walkLeft( );
41                        mode = 4;
42                    }
43                    else{

```

```

44         value = target.walkDown( );
45     }
46     break;
47     case 4: /* 壁にぶつかるまで左に移動する */
48         if(value[4] == 2){
49             value = target.walkUp( );
50             mode = 1;
51         }
52         else{
53             value = target.walkLeft( );
54         }
55         break;
56     }
57
58     /****** 制御情報が 0 だったら終了する *****/
59     if(value[0] == 0) break;
60 }
61
62 /****** 競技サーバから切断する *****/
63 target.exit( );
64 }
65 }
66

```

## プログラム説明

### ①変数の宣言

```

4     int[ ] value;
5     value = new int[10];
6
7     int mode = 1;

```

**value** は、サーバからの戻り値（制御情報 1 + 周囲情報 9）を格納する配列です。

**mode** は、動作の種類を記憶させる変数です。**mode** を使うことで、どの動作をしているのかを管理することができます。サンプルプログラムは下表のように 4 つのモードを用意しています。この方式ならば、今後プログラムを発展させるときモードを追加するだけで様々な動作ができるようになります。

### ・モードの種類と動作

| 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 |

## ②競技サーバに接続する

```

9      /***** 競技サーバに接続する *****/
10     edu.procon.PreConnect2008 target;
11     target = new edu.procon.PreConnect2008( );

```

edu.procon.PreConnect2008 クラスを使って競技サーバに接続します。

## ③getReady でサーバからの戻り値を得る

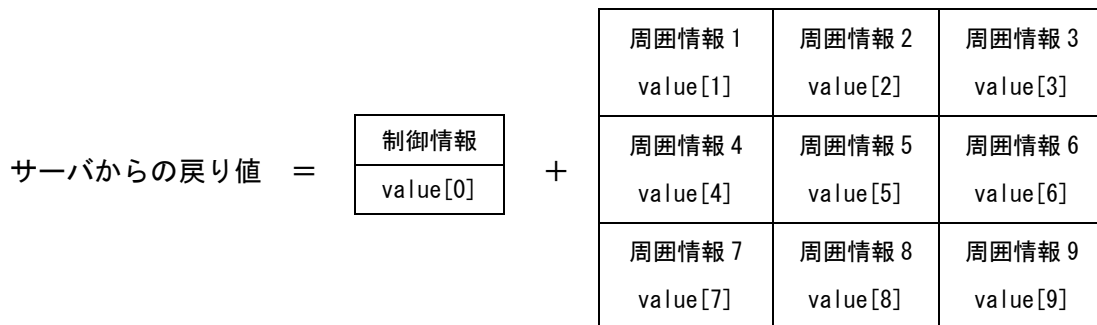
```

14     /***** まず、getReady でサーバから戻り値を得る *****/
15     value = target.getReady( );
16     if(value[0] == 0) break;

```

getReady メソッドを送信するとサーバから戻り値が返ってくるので、配列 value で受け取ります。value[0]に制御情報が格納されているので、0 だったら終了します。

・ 配列 value の内容（サーバからの戻り値）



## ④mode で分岐する

```

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                 value = target.walkRight( ); /* 右に移動し */
23                 mode = 2; /* mode を 2 に変更する */
24             }
25             else{
26                 value = target.walkUp( ); /* そうでなければ、上に移動する */
27             }
28             break;
29         case 2: /* 壁にぶつかるまで右に移動する */
30             .
31             .
32             .

```

switch 文を使って mode の値で分岐処理を行います。mode = 1 の場合、壁にぶつかるまで上に移動するので、自分のすぐ上が壁かどうかを if(value[2] == 2) で判断します。

もし、上が壁だった場合は次に右に移動するので walkRight メソッドを送信し、mode を 2 にします。壁でなければ、walkUp メソッドを送信し上に移動します。

同様に mode2 から 4 を作ります。

## ⑤制御情報が 0 だったら終了する

```
58      /***** 制御情報が 0 だったら終了する *****/  
59      if(value[0] == 0) break;
```

メソッドを送信するとサーバから戻り値が返ってくるので、終了の判定をします。

## ⑥競技サーバから切断する

```
62      /***** 競技サーバから切断する *****/  
63      target.exit( );
```

exit でメソッドで競技サーバから切断します。

**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。