

全国高校生プログラミングコンテスト

CHaserOnline

ステップアップヒント2

⑧ コマンドについて

コマンドについて説明します。CやHなどのプレイヤー（以下プレイヤー）の制御をおこなうにはコマンドを使用します。コマンドには以下のものがあります（2014.6.12 現在）。説明中のプレイヤーはCと表記します。また、戻り値が入る変数は整数型の一次元配列 `returnNumber[10]` とします。





一連のコマンド発行後に「user=」メッセージを受け取った場合はゲーム終了となりますので、クライアントプログラムを終了してください。





A 準備コマンド（サーバに接続し、プレイヤーの周囲情報を得る）										
コマンド名	機能									
gr	サーバに接続し、自分のターンであればプレイヤーの周囲情報を得る。									
	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>C</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	C	5	6	7	8
	0	1	2							
	3	C	5							
6	7	8								
マス目にある数字は <code>returnNumber[]</code> の添え字と一致する。										

B 動作コマンド walk 系 (指定した方向へ 1 マス移動する)																									
コマンド名	機能																								
wu wd wl wr	<p>指定した方向へ 1 マス移動する。</p> <p>wu・・・上 wd・・・下 wl・・・左 wr・・・右</p> <p>wr の例</p> <table border="1"> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td></td> <td>3</td> <td>→ C</td> <td>5</td> </tr> <tr> <td></td> <td>6</td> <td>7</td> <td>8</td> </tr> </table> <p>マス目にある数字は returnNumber[] の添え字と一致する。 また、数字の並びは wu, wd, wl の場合も移動先の周囲 9 マスの左上が 0 となる。</p> <p>wl の例</p> <table border="1"> <tr> <td>0</td> <td>1</td> <td>2</td> <td></td> </tr> <tr> <td>3</td> <td>C ←</td> <td>5</td> <td></td> </tr> <tr> <td>6</td> <td>7</td> <td>8</td> <td></td> </tr> </table>		0	1	2		3	→ C	5		6	7	8	0	1	2		3	C ←	5		6	7	8	
	0	1	2																						
	3	→ C	5																						
	6	7	8																						
0	1	2																							
3	C ←	5																							
6	7	8																							

B 動作コマンド look 系 (指定した方向の周囲 9 マスの情報を得る)																															
コマンド名	機能																														
lu	指定した方向の周囲 9 マスの情報を得る。 lu・・・上 ld・・・下 ll・・・左 lr・・・右 lr の例 <table border="1" data-bbox="411 618 975 972"> <tr><td></td><td></td><td>0</td><td>1</td><td>2</td></tr> <tr><td></td><td>C</td><td>3</td><td>4</td><td>5</td></tr> <tr><td></td><td></td><td>6</td><td>7</td><td>8</td></tr> </table> <p>マス目にある数字は returnNumber[] の添え字と一致する。 また、数字の並びは lu, ld, ll の場合も探査範囲の左上が 0 となる。</p> ld の例 <table border="1" data-bbox="411 1167 759 1753"> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>C</td><td></td></tr> <tr><td>0</td><td>1</td><td>2</td></tr> <tr><td>3</td><td>4</td><td>5</td></tr> <tr><td>6</td><td>7</td><td>8</td></tr> </table>			0	1	2		C	3	4	5			6	7	8					C		0	1	2	3	4	5	6	7	8
			0	1	2																										
		C	3	4	5																										
			6	7	8																										
	C																														
0	1	2																													
3	4	5																													
6	7	8																													
ld																															
ll																															
lr																															

B 動作コマンド search系 (指定した方向、真っすぐ9マスの情報を得る)																																																																																																				
コマンド名	機能																																																																																																			
su	指定した方向、真っすぐ9マスの情報を得る。																																																																																																			
sd	su・・・上																																																																																																			
sl	sd・・・下																																																																																																			
sr	sl・・・左 sr・・・右																																																																																																			
	<p>sr の例</p> <table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>C</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table> <p>マス目にある数字は returnNumber[] の添え字と一致する。 また、数字の並びは左上のマスが 0 となる。</p> <p>sd の例</p> <table border="1"> <tr><td></td><td></td><td></td></tr> <tr><td></td><td>C</td><td></td></tr> <tr><td></td><td>0</td><td></td></tr> <tr><td></td><td>1</td><td></td></tr> <tr><td></td><td>2</td><td></td></tr> <tr><td></td><td>3</td><td></td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> </table> <p>su の例</p> <table border="1"> <tr><td></td><td>0</td><td></td></tr> <tr><td></td><td>1</td><td></td></tr> <tr><td></td><td>2</td><td></td></tr> <tr><td></td><td>3</td><td></td></tr> <tr><td></td><td>4</td><td></td></tr> <tr><td></td><td>5</td><td></td></tr> <tr><td></td><td>6</td><td></td></tr> <tr><td></td><td>7</td><td></td></tr> <tr><td></td><td>8</td><td></td></tr> <tr><td></td><td>C</td><td></td></tr> <tr><td></td><td></td><td></td></tr> </table>													C	0	1	2	3	4	5	6	7	8																C			0			1			2			3			4			5			6			7			8			0			1			2			3			4			5			6			7			8			C				
	C	0	1	2	3	4	5	6	7	8																																																																																										
	C																																																																																																			
	0																																																																																																			
	1																																																																																																			
	2																																																																																																			
	3																																																																																																			
	4																																																																																																			
	5																																																																																																			
	6																																																																																																			
	7																																																																																																			
	8																																																																																																			
	0																																																																																																			
	1																																																																																																			
	2																																																																																																			
	3																																																																																																			
	4																																																																																																			
	5																																																																																																			
	6																																																																																																			
	7																																																																																																			
	8																																																																																																			
	C																																																																																																			

B 動作コマンド put 系（指定した方向へブロックを置く）										
コマンド名	機能									
pu2 pd2 pl2 pr2	<p>指定した方向へブロックを置く。</p> <p>pu2・・・上 pd2・・・下 pl2・・・左 pr2・・・右</p> <p>pr2 の例</p> <table border="1"> <tr> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td>3</td> <td>C →</td> <td></td> </tr> <tr> <td>6</td> <td>7</td> <td>8</td> </tr> </table> <p>※  は壁</p> <p>マス目にある数字は returnNumber[] の添え字と一致する。</p>	0	1	2	3	C →		6	7	8
0	1	2								
3	C →									
6	7	8								

B 動作コマンド put&walk 系 (指定した方向へブロックを置き、逆向きに1マス移動する)																	
コマンド名	機能																
pu2wd	指定した方向へブロックを置き、逆向きに1マス移動する。																
pd2wu	pu2wd・・・上にブロックを置き、下へ移動																
pl2wr	pd2wu・・・下にブロックを置き、上へ移動																
pr2wl	pl2wr・・・左にブロックを置き、右へ移動																
pru2wld	pr2wl・・・右にブロックを置き、左へ移動																
plu2wrd	pru2wld・・・右上にブロックを置き、左下へ移動																
prd2wlu	plu2wrd・・・左上にブロックを置き、右下へ移動																
pld2wru	prd2wlu・・・右下にブロックを置き、左上へ移動 pld2wru・・・左下にブロックを置き、右上へ移動																
	<p>pld2wru の例</p> <table border="1"> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td></td> <td>3</td> <td>C</td> <td>5</td> </tr> <tr> <td></td> <td>6</td> <td>7</td> <td>8</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p>※  は壁</p> <p>マス目にある数字は returnNumber[] の添え字と一致する。</p>		0	1	2		3	C	5		6	7	8				
	0	1	2														
	3	C	5														
	6	7	8														
																	

B 動作コマンド kei 系 (桂馬に似た動き)																					
コマンド名	機能																				
keiru	指定した方向へ桂馬に似た動きをする。																				
keird	keiru・・・右上へ桂馬のような動きをする																				
keilu	keird・・・右下へ桂馬のような動きをする																				
keild	keilu・・・左上へ桂馬のような動きをする keild・・・左下へ桂馬のような動きをする																				
	<p>keird の例</p> <table border="1"> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> </tr> <tr> <td></td> <td>3</td> <td>C</td> <td>5</td> </tr> <tr> <td></td> <td>6</td> <td>7</td> <td>8</td> </tr> </tbody> </table> <p>マス目にある数字は returnNumber[] の添え字と一致する。</p>										0	1	2		3	C	5		6	7	8
	0	1	2																		
	3	C	5																		
	6	7	8																		

C 終了コマンド (自分のターンを終了させる)	
コマンド名	機能
#	自分のターンを終了させる ※周囲情報なし

9 プログラムの流れについて

みなさんが作るクライアントプログラムと競技サーバのやりとりは次のようになります。

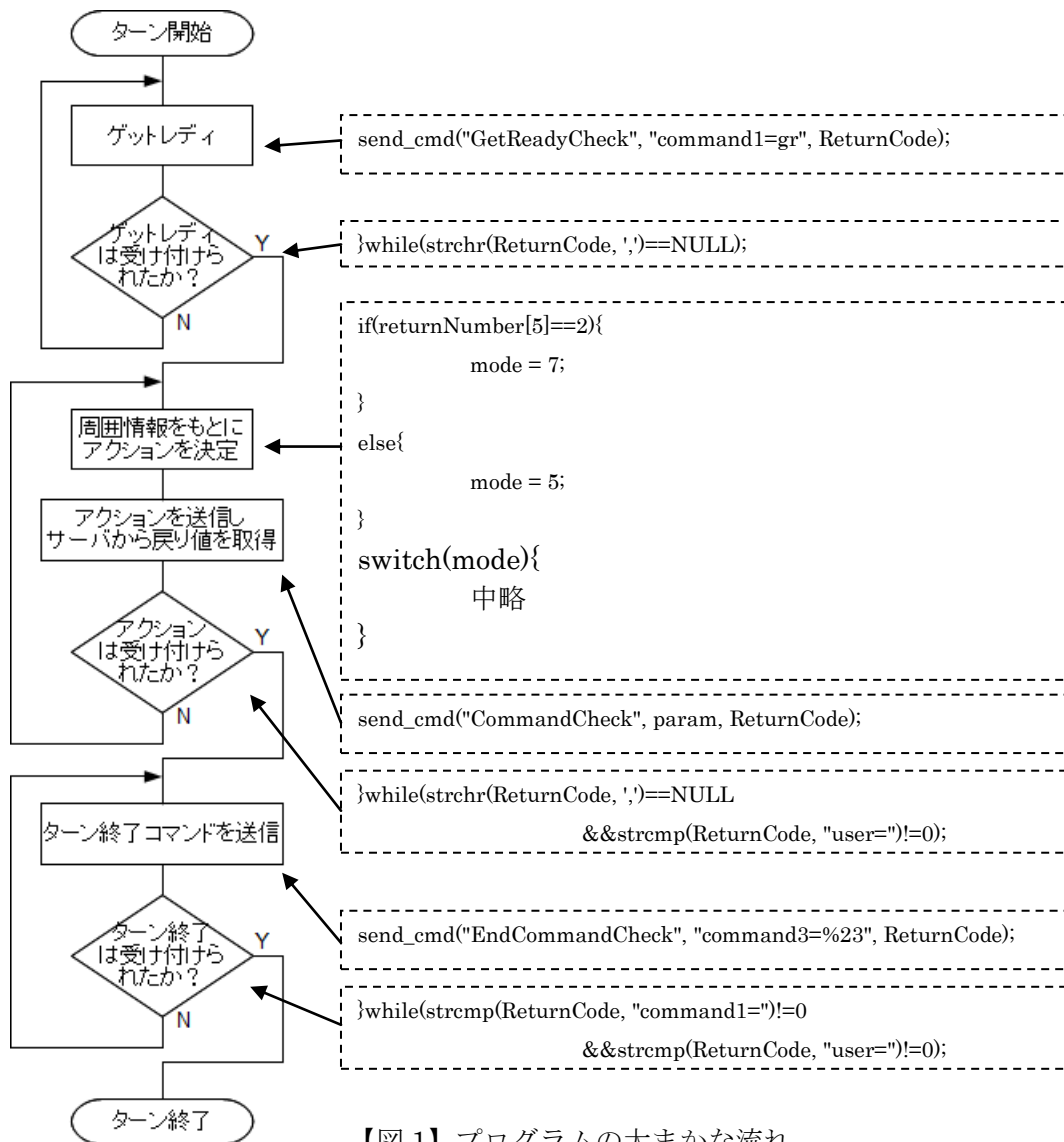
まず WEB の仕組みとしてサーバへの接続、WEB コマンドの送信、サーバからのリターンメッセージの受信とサーバからの切断がひとつのセットとなります。複数のセットを連携させる仕組みとして「セッション」を利用します。

準備コマンドの `gr` をサーバへ送信して周囲情報を取得するセット、動作コマンドを選んで送信して周囲情報を取得するセット、ターンを終了させるセットの3つのセットをひとつのパッケージとします。このパッケージをターン終了まで繰り返します。クライアントプログラムは図1のフローチャートのようになります。

みなさんは、サーバから得た周囲情報をもとにコマンドを選んで送信する部分を考えます。

【フローチャート】

【プログラム】(途中省略あり)



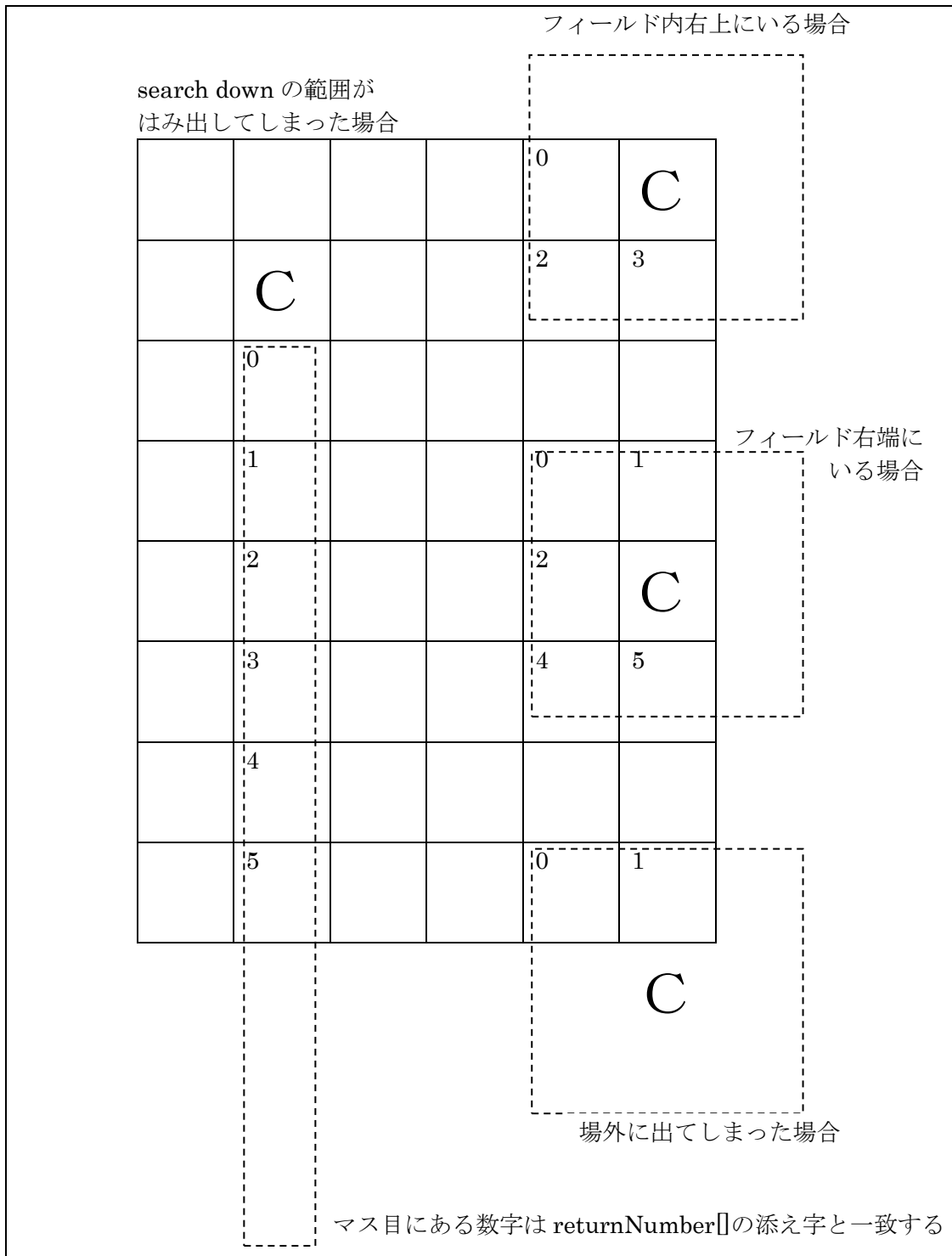
【図1】プログラムの大まかな流れ

10 サンプルプログラム 2 (フィールド端を捉える)

① 場外について

クライアントがフィールド外に近づく、または出てしまうと周囲情報が減少します。この仕組みを利用してフィールド外へ出ないようにします。

フィールド外に近づいた場合の例



② プログラムの保存

サンプルプログラム2をダウンロードするか、アクション部分を打ち直します。
プログラム名は「CHaserOnlineClient009-3FieldRelease.c」です。

③ プログラムの動作

- ・まず右へ移動します。
- ・右端までたどり着いたら下へ移動します。
- ・右下までたどり着いたら左へ移動します。
- ・左下までたどり着いたら上へ移動します。
- ・左上までたどり着いたら右へ移動します。以降はフィールドの内側を時計回りに制限タ
ーンになるまで移動を繰り返します。
- ・mode(モード)という変数を作り、動作の種類を記憶させています。

④ 対戦(1台のパソコンで二つのクライアントを起動する場合)

- ・端末を二つ起動させる。
 - ・二つの画面でそれぞれのコマンドを入力し、対戦させる。
- ひとつは自分のIDで起動させ、もうひとつはcoolやhotなどの公開されているIDで起
動する。

起動コマンドの例(自分のID)

```
./CHaserOnlineClient009-3FieldRelease.o http://www7019ug.sakura.ne.jp:80/  
CHaserOnline003/user/_-u_uJibun_-p_Watashi_-r_1_-x_192.168.30.251:8080
```

起動コマンドの例(もうひとつのID)

```
./CHaserOnlineClient009-3FieldRelease.o http://www7019ug.sakura.ne.jp:80/  
CHaserOnline003/user/_-u_ucool_-p_cool_-r_1_-x_192.168.30.251:8080
```

※先に接続したクライアントから順にC、Hのキャラクターが割振られます。また、戻り値
はそれぞれ1000、2000となります。

⑤ サンプルプログラム

```
/*-----  
Action を発行する  
-----*/  
do{  
    if(beforeCount != count){  
        if(beforeCount > count){  
            switch(mode){  
                case 1:           //上なら  
                    mode = 5;     //右へ  
                    break;  
  
                case 5:           //右なら  
                    mode = 7;     //下へ  
                    break;  
  
                case 7:           //下なら  
                    mode = 3;     //左へ  
                    break;  
  
                case 3:           //左なら  
                    mode = 1;     //上へ  
                    break;  
  
                default:          //それ以外なら  
                    mode = 5;     //右へ  
            }  
        }  
        beforeCount = count;  
    }  
    strcpy(param, "command2=");  
    switch(mode){  
        中略  
    }  
    send_cmd("CommandCheck", param, ReturnCode);  
}while(strchr(ReturnCode, ',')==NULL&&strcmp(ReturnCode,"user=")!=0);
```

⑥ プログラム説明

```
if(beforeCount != count){  
    .  
    .  
    .  
    beforeCount = count;  
}
```

前回の戻り値の数と変化があったら処理をします。
変化があった場合は次回のために戻り値の数を記録します。

```
if(beforeCount > count){  
    switch(mode){  
        .  
        .  
        .  
    }  
}
```

もしも、前回よりも戻り値の数が減少した場合は処理をします。

```
switch(mode){  
    case 1:           //上なら  
        mode = 5;    //右へ  
        break;  
    .  
    .  
    .  
}
```

右回りになるように方向を変化させます。

上記の処理を組み合わせることで、フィールド端を察知して場外へ出なくすることができます。

⑦ 注意

このクライアントは一度フィールド端を見つけると、ターン終了までフィールド端を右回りになぞり続けます。進行方向に壁があってもよけることなく進みますので、減点が増えてしまいます。うまく回避できるように判断を追加してみてください。

11 新アイテム、新コマンドについて

ワープや **put&walk** 系コマンド、**kei** 系コマンドについて、使い方のヒントや注意点について説明します。

まずワープについて説明します。

このワープはコマンドではなくアイテム類であることに注意してください。

ステップアップヒント1の8ページでも触れましたが、このワープを取ろうとすると上下左右10マス分移動することができます。左右への移動で考えると **walk** や **put&walk** の10ターン分を1ターンで移動することができます。

違うエリアに移動したいときに使うと良いでしょう。ただし、移動先に他クライアントがいる可能性も考えられますので注意が必要となります。

次に **put&walk** 系コマンドについて説明します。

このコマンドは1ターンで **put** と **walk** の二つの動作を組み合わせて実行できる効率のよいコマンドです。**CHaserOnline** では相手に **Put** をしてもゲームは終了せずにターン終了まで進みます。相手に **Put** したらその逆に動くことができるこのコマンドは活用の場が多くありそうです。また、上下左右の動きに加えて斜め方向にも動くことができますので **walk** のみでの移動よりも早く動くことができます。

しかし必ず **Put** してから、その逆方向に動きますのでアイテムをつぶしてしまう危険性もあります。

最後に **kei** 系コマンドについて説明します。

このコマンドは **walk** の3ターン分を1ターンで移動することができます。うまく活用すれば少ないターン数で広範囲の移動や探査が可能になります。ブロックに囲まれてしまい身動きがとれないときにも効果を発揮するでしょう。しかし移動先はゲットレディでは見えない場所ですので注意が必要です。