

```

1
2 import java.util.InputMismatchException;
3
4
5 public class New_profile {
6     //profile入力
7
8     public static void main(String[] args) {
9
10        //変数初期化处理
11        final String ID = "ID";
12        final String NAMAЕ = "名前";
13        final String SEINENGAPPI = "生年月日";
14        final String KETUEKIGATA = "血液型";
15        final int KOUMOKUSU = 3; //IDを含めないstatusの数
16        final String NYUURYOKUSAKI [] = {ID, NAMAЕ, SEINENGAPPI, KETUEKIGATA}; //ステータス名を配列に保存
17        int okonauDousa = 0; //現在の状態。
18        int genzaiKoumoku = 0; //編集中のstatus
19        String profID [] = new String[KOUMOKUSU+1]; //格納先 [0]=ID、[1~]値
20        boolean owari = true; //プログラム終了の判定
21        boolean syokai = true; //初回登録の判定。
22        boolean hensyuOwari = true; //編集終わりの判定
23
24        //条件の指定
25        int yobidashiID = 0; //今回編集するIDの人
26        //今回はID=0の人のプロフィール入力指定
27        syokai = true; //初回オン
28        //今回は初回指定
29
30        do {
31            if(syokai) { //初回、もしくはクリアの時の処理
32
33                String yobidashiIDmozi = String.valueOf(yobidashiID); //IDを文字列に変換
34
35                genzaiKoumoku = 0; //入力中のステータスをIDに変更
36                profID[genzaiKoumoku] = yobidashiIDmozi; //入力するIDの読み込み
37                System.out.println("ID:" + profID[genzaiKoumoku] + "のprofile入力を開始します");
38                okonauDousa = 2; //状態をステータス編集に指定
39                genzaiKoumoku = 1; //入力ステータスを初期に遷移
40
41            } else { //初期でない場合次の動作を確認
42
43                genzaiKoumoku = 1; //入力中のステータス番号初期化
44                System.out.println("ID:" + profID[0] + "のprofileの情報を確認します");
45
46                for(int i = genzaiKoumoku; i <= KOUMOKUSU; i++) { //ステータスを一覧表示するまで繰り返す
47                    System.out.println(NYUURYOKUSAKI[i] + ":" + profID[i]);
48                } //繰り返し終了
49
50                System.out.println("以上の内容でよろしいですか?");
51                System.out.println("1:確定 2:編集 3:最初からやりなおす");
52                int zyougen = 3;
53                //選択の上限を格納
54
55                okonauDousa = Input_SuziNyuryoku(zyougen); //選択の上限を引数に指定
56            }
57
58            switch(okonauDousa) { //選択によって分岐
59
60                case 1:
61                    //確定・保存
62                    System.out.println("入力内容を保存します");
63                    Hozon();
64                    owari = false;
65
66                case 2:
67                    //ステータスの編集
68
69                    if(syokai == false) {
70                        //初回でない場合、ステータス一覧を表示し編集するステータスを選ばせる
71
72                        System.out.println("編集するステータスを選んでください");
73
74                        for(int i = 1; i <= KOUMOKUSU; i++) { //IDを除いたステータス一覧を表示するまで繰り返す
75                            System.out.println(i + ":" + NYUURYOKUSAKI[i]);
76                        } //繰り返し終了
77
78                        genzaiKoumoku = Input_SuziNyuryoku(KOUMOKUSU); //入力された数字を入力中にする //選択の上
79                        //限を引数に指定
80                        System.out.println(NYUURYOKUSAKI[genzaiKoumoku] + "を編集します");
81                    } //if終了

```

```
82
83     do {
84         //初回の場合は項目が埋まるまで繰り返す
85
86         boolean i = true; //SWの用意
87         do { //ステータスを確定するまで繰り返す
88
89             //ステータス入力
90             String nyuuryoku = KOUMOKUSUNyuuryoku (genzaiKoumoku, NYUURYOKUSAKI [genzaiKoumoku]);
91             //入力処理
92
93             String YNshitumon = NYUURYOKUSAKI[genzaiKoumoku] + “は「 + nyuuryoku + ”でよろしい
94     ですか?”;
95             //YN質問に格納
96
97             if(“Y”.equals(Input_YorN (YNshitumon).toUpperCase())) { //YN質問メソッドと回答
98                 //Yならステータス入力を抜ける
99                 i = false;
100                //抜け出す切り替え
101                profID[genzaiKoumoku] = nyuuryoku; //格納
102            } //if終了
103
104        } while (i); //ステータス入力終了
105
106
107        if (genzaiKoumoku >= KOUMOKUSU || syokai == false) { //編集を抜ける条件
108            syokai = false;
109            hensyuOwari = false;
110        }
111
112        if (syokai) {
113            genzaiKoumoku += 1;
114            //初回用にactiveをずらす
115        }
116
117        } while (hensyuOwari);
118
119
120    case 3:
121        //やりなおし
122        //全ステータスを初期化する処理入れたい、未入力
123
124    } //Switch終了
125
126
127    } while (owari);
128
129    System.out.println(“complete^^”);
130 } //Mainメソッド終了
131
132 public static String KOUMOKUSUNyuuryoku(int i, String NYUURYOKUSAKI) {
133
134
135    System.out.println(NYUURYOKUSAKI + “を入力してください”);
136    //入力指示
137
138    Scanner scan = new Scanner (System.in);
139    //Scannerを呼び出し
140    String nyuuryoku = scan.next();
141    //入力した値の格納
142
143
144    return nyuuryoku;
145 }
146
147
148 public static void Hozon() {
149    //保存の処理
150
151    System.out.println(“保存終了”);
152    //終了
153 }
154
155 public static String Input_YorN(String shitumon) {
156
157    System.out.println(shitumon);
158    //入力させる文言
159
160    String YN = null;
161    boolean i = true;
```

```
162     //変数初期化
163
164     do {
165         System.out.print("Yes/No");
166
167         Scanner scan = new Scanner(System.in);
168         //Scannerを呼び出し
169         YN = scan.next();
170         //読み取ったものをYNIに入れるf
171
172         if(("Y".equals(YN.toUpperCase()) || ("N".equals(YN.toUpperCase())) {
173             //YまたはNが一致なら抜ける。大文字小文字対応。
174             i = false;
175         }
176     }while (i);
177
178     return YN;
179 }
180
181 }
182
183 public static int Input_SuziNyuryoku(int zyougen) {
184
185     int Suuzi = 0;
186     int i = 0;
187     //変数とスイッチの初期化
188
189     do {
190         Scanner scan = new Scanner(System.in);
191         //Scannerを呼び出し
192
193         try {
194             Suuzi = scan.nextInt();
195             //読み取ったものをSuuziに入れる
196         } catch (InputMismatchException e) {
197
198             System.out.println("半角数字を入力してください");
199             //エラー処理
200         }
201
202         if(Suuzi <= zyougen && !(0 == Suuzi)) {
203             //数字が正しければ抜ける
204             i = 1;
205         } else {
206             System.out.println(zyougen + "以内で入力してください");
207         }
208     }while (i == 0);
209
210     return Suuzi;
211 }
212
213 }
214
215 }
216
217 }
218
```