```
 1
 2 package utility;
 3
 4 import java.util.Scanner;
 5
 6 /**
 7  *
 8  * @author Chew Lip Sin
 9  */
10 public class InputValue {
11
12     Scanner sc = new Scanner(System.in);
13     private final static Scanner in = new Scanner(System.in);
14
15     public int readInteger() {
16         int number = 0;
17         boolean isInteger = false;
18         do {
19             isInteger = false;
20             try {
21                 number = sc.nextInt();
22                 sc.nextLine();
23             } catch (Exception e) {
24                 MessageUI.printFormattedText("Invalid format, please enter in integer: ",
ConsoleColor.RED);
25                 sc.next();
26                 isInteger = true;
27             }
28
29         } while (isInteger);
30         return number;
31     }
32
33     public String readString() {
34         String input;
35         do {
36             input = sc.nextLine();
37             input = input.trim();
38             if (input.equals("")) {
39                 MessageUI.displayInvalidFormat();
40             }
41         } while (input.equals(""));
42         return input;
43     }
44
45     public String readCourseCode() {
```

```
 46          String input;
 47          boolean match = false;
 48          do {
 49              input = sc.nextLine();
 50
 51              match =
input.matches("^[ABFMabfm][A-Za-z]{3}\\d{4}$|^[Mm][Pp][Uu][-]\\d{4}$");
 52              if ("0".equals(input)) {
 53                  match = true;
 54                  return "0";
 55              }
 56              if (!match) {
 57                  MessageUI.displayInvalidFormat();
 58              }
 59          } while (!match);
 60          return input;
 61      }
 62
 63      public static int inputInt(String prompt) {
 64          boolean success = false;
 65          int result = 0;
 66          do {
 67              try {
 68                  System.out.print(prompt);
 69                  result = in.nextInt();
 70                  in.nextLine();
 71                  success = true;
 72              } catch (Exception ex) {
 73                  in.nextLine();
 74                  System.out.println("** Input is not a number, try again. **");
 75              }
 76          } while (!success);
 77          return result;
 78      }
 79
 80      public static int inputChoice(String prompt, int min, int max) {
 81          int input;
 82          while (true) {
 83              try {
 84                  System.out.print(prompt);
 85                  String inputStr = in.nextLine().trim();
 86
 87                  if (inputStr.isEmpty()) {
 88                      return 8;
 89                  }
 90
```

```
 91                 input = Integer.parseInt(inputStr);
 92
 93                 if (input >= min && input <= max) {
 94                     break;
 95                 } else {
 96                     System.out.println("Invalid input. Please enter a number between " +
min + " and " + max + ".");
 97                 }
 98             } catch (NumberFormatException ex) {
 99                 System.out.println("Invalid input. Please enter a valid number.");
100             }
101         }
102         return input;
103     }
104
105     public static int intChoice(String prompt, int min, int max) {
106         int input;
107         while (true) {
108             try {
109                 System.out.print(prompt);
110                 String inputStr = in.nextLine().trim();
111
112                 if (inputStr.isEmpty()) {
113                     System.out.println("Input cannot be empty. Please enter a number
between " + min + " and " + max + ".");
114                     continue;
115                 }
116
117                 input = Integer.parseInt(inputStr);
118
119                 if (input >= min && input <= max) {
120                     break;
121                 } else {
122                     System.out.println("Invalid input. Please enter a number between " +
min + " and " + max + ".");
123                 }
124             } catch (NumberFormatException ex) {
125                 System.out.println("** Input is not a number, try again. **");
126             }
127         }
128         return input;
129     }
130
131     public static String inputString(String prompt) {
132         System.out.print(prompt);
133         String input = in.nextLine();
```

```
134         return input;
135     }
136 
137     public static String enterString(String prompt) {
138         String input;
139         do {
140             System.out.print(prompt);
141             input = in.nextLine();
142             if (input.trim().isEmpty()) {
143                 System.out.println("** Input cannot be blank. Please try again. **");
144             }
145         } while (input.trim().isEmpty());
146         return input;
147     }
148 
149     public static char inputChar(String prompt) {
150         String input;
151         do {
152             System.out.print(prompt);
153             input = in.nextLine().trim();
154             if (input.trim().isEmpty()) {
155                 System.out.println("** Input cannot be blank. Please try again. **");
156             }
157         } while (input.trim().isEmpty());
158         return input.charAt(0);
159     }
160 
161     public static char inputYN() {
162         char inputYN = 0;
163         boolean success = false;
164 
165         do {
166             String input = in.nextLine();
167 
168             if (input.length() == 1 && (input.charAt(0) == 'y' || input.charAt(0) ==
'n')) {
169                 inputYN = input.charAt(0);
170                 success = true;
171             } else {
172                 System.out.println("Invalid input. Please enter 'y' or 'n'.");
173             }
174         } while (!success);
175 
176         return inputYN;
177     }
178 
```

```
179     public static void closeInput() {
180         in.close();
181     }
182 
183     public boolean isValidCode(String code) {
184         return code.matches("[A-Za-z]{3}");
185     }
186 
187     public boolean isValidTutorialGroupName(String groupName) {
188         return groupName.matches("[A-Za-z]{3}G([1-9]|1\\d|2[0-9]|30)"); // 2[0-9]
matches any two-digit number starting with 2 (20 to 29).
189         //followed by a number from 1 to 30
190     }
191 //    public String readAlphaInt() {
192 //        String input;
193 //        boolean matchs;
194 //        String regex = "\\dA-Za-z\\s-]+";
195 //        do {
196 //        input = sc.nextLine();
197 //        matchs = input.matches(regex);
198 //
199 //        if ("0".equals(input)) {
200 //            matchs = true;
201 //            return input;
202 //        }
203 //
204 //            if (!matchs) {
205 //                MessageUI.displayInvalidFormat();
206 //            }
207 //        } while (!matchs);
208 //        return input;
209 //    }
210 }
```