```java
package application;

import java.util.Date;


public class CommonHealthData extends HealthData {
    private String metric;
    private int systolicBP;
    private int diastolicBP;
    private int age;
    private double weight;
    private double height;
    private int ldlCholesterol;
    private int hdlCholesterol;
    private int triglycerideCholesterol;
    private double glucoseLevel;

    public CommonHealthData(String name, Date date, String metric, int systolicBP, int diastolicBP) {
        super(name, date);
        this.metric = metric;
        this.systolicBP = systolicBP;
        this.diastolicBP = diastolicBP;
    }

    public CommonHealthData(String name, Date date, String metric, int ldlCholesterol, int hdlCholesterol, int triglycerideCholesterol) {
        super(name, date);
        this.metric = metric;
        this.ldlCholesterol = ldlCholesterol;
        this.hdlCholesterol = hdlCholesterol;
        this.triglycerideCholesterol = triglycerideCholesterol;
    }


    public CommonHealthData(String name, Date date, String metric, double glucoseLevel) {
        super(name, date);
        this.metric = metric;
        this.glucoseLevel = glucoseLevel;
    }

    public CommonHealthData(String name, Date date, String metric, double weight, double height) {
        super(name, date);
        this.metric = metric;
        this.weight = weight;
        this.height = height;
    }


    @Override
    public String getMetric() {
        return metric;
    }

    public int getSystolicBP() {
        return systolicBP;
    }
```

```
57
58      public int getDiastolicBP() {
59          return diastolicBP;
60      }
61
62      public int getAge() {
63          return age;
64      }
65      public void setSystolicBP(int systolicBP) {
66          this.systolicBP = systolicBP;
67      }
68
69      public void setDiastolicBP(int diastolicBP) {
70          this.diastolicBP = diastolicBP;
71      }
72      public void setAge(int age) {
73          this.age = age;
74      }
75
76      public double getWeight() {
77          return weight;
78      }
79
80      public void setWeight(double weight) {
81          this.weight = weight;
82      }
83
84      public double getHeight() {
85          return height;
86      }
87
88      public void setHeight(double height) {
89          this.height = height;
90      }
91
92      public int getLdlCholesterol() {
93          return ldlCholesterol;
94      }
95
96      public void setLdlCholesterol(int ldlCholesterol) {
97          this.ldlCholesterol = ldlCholesterol;
98      }
99
100     public int getHdlCholesterol() {
101         return hdlCholesterol;
102     }
103
104     public void setHdlCholesterol(int hdlCholesterol) {
105         this.hdlCholesterol = hdlCholesterol;
106     }
107
108     public int getTriglycerideCholesterol() {
109         return triglycerideCholesterol;
110     }
111
112     public void setTriglycerideCholesterol(int triglycerideCholesterol) {
113         this.triglycerideCholesterol = triglycerideCholesterol;
114     }
115
```

```
116     public double getGlucoseLevel() {
117         return glucoseLevel;
118     }
119 
120     public void setGlucoseLevel(double glucoseLevel) {
121         this.glucoseLevel = glucoseLevel;
122     }
123 
124     public double calculateBMI() {
125         /**
126          * Calculates the Body Mass Index (BMI) based on the height and weight values.
127          *
128          * Pre-condition:
129          * - The height and weight values have been set.
130          *
131          * Post-condition:
132          * - The BMI value is calculated and returned.
133          * - The height and weight values remain unchanged.
134          */
135         double heightInMeters = height * 0.0254; // Convert inches to meters (1 inch =
    0.0254 meters)
136         double weightInKilograms = weight * 0.453592; // Convert pounds to kilograms
    (1 pound = 0.453592 kilograms)
137         double bmi = weightInKilograms / (heightInMeters * heightInMeters);
138         DecimalFormat decimalFormat = new DecimalFormat("#.##"); // Format to two
    decimal places
139         return Double.parseDouble(decimalFormat.format(bmi));
140     }
141     public void validate() throws HealthDataException {
142         /**
143          * Validates the health data based on the metric type.
144          *
145          * @throws HealthDataException if the health data fails validation
146          *
147          * Pre-condition:
148          * - The health data values have been set.
149          *
150          * Post-condition:
151          * - The health data is valid/positive according to the specific metric type.
152          * - If the validation fails, a HealthDataException is thrown.
153          */
154         if (systolicBP < 0 || diastolicBP < 0) {
155             throw new HealthDataException("Blood pressure values cannot be
    negative.");
156         }
157         if (getMetric().equals("BMI")) {
158         if (weight <= 0 || height <= 0) {
159             throw new HealthDataException("Weight and height values must be
    positive.");
160         }
161         }
162         if (getMetric().equals("Cholesterol")) {
163             if (ldlCholesterol < 0 || hdlCholesterol < 0 || triglycerideCholesterol <
    0) {
164                 throw new HealthDataException("Cholesterol values must be positive.");
165             }
166         }
167         if (getMetric().equals("Blood Glucose")) {
168         if ( glucoseLevel <= 0) {
```

```
169             throw new HealthDataException("glucoseLevel value must be positive.");
170         }
171         }
172     }
173
174
175
176     @Override
177     public String getData() {
178         StringBuilder data = new StringBuilder();
179         data.append("Recorded at: ").append(getDate()).append("\n");
180         data.append("Metric: ").append(metric).append("\n");
181         data.append("Systolic BP: ").append(systolicBP).append("\n");
182         data.append("Diastolic BP: ").append(diastolicBP).append("\n");
183         data.append("BMI: ").append(calculateBMI()).append("\n");
184         if (age != 0) {
185             data.append("Age: ").append(age).append("\n");
186         }
187         if (weight != 0) {
188             data.append("Weight: ").append(weight).append("\n");
189         }
190         if (height != 0) {
191             data.append("Height: ").append(height).append("\n");
192         }
193         if (ldlCholesterol != 0) {
194             data.append("LDL Cholesterol: ").append(ldlCholesterol).append("\n");
195         }
196         if (hdlCholesterol != 0) {
197             data.append("HDL Cholesterol: ").append(hdlCholesterol).append("\n");
198         }
199         if (triglycerideCholesterol != 0) {
200             data.append("Triglyceride Cholesterol: ").append
    (triglycerideCholesterol).append("\n");
201         }
202         if (glucoseLevel != 0) {
203             data.append("Glucose Level: ").append(glucoseLevel).append("\n");
204         }
205         return data.toString();
206     }
207 }
208
```