The Alert Generation System

The Alert Generation System is designed to actively manage patient-specific medical alerts within a cardiovascular ward in a hospital.

Central Classes:

The **"AlertGenerator"** class is responsible for continuous evaluation of data incoming from various medical devices and initializing action, when thresholds are crossed.
It has two methods: evaluateData() and triggerAlert(). Separation of data monitoring and alert triggering ensures clear distinction between two tasks of the class, providing clarity and maintainability.

The **"Alert"** class captures all necessary information about an alert within four fields: patientId, alertId, timestamp and condition. By providing detailed information about each alert, the class ensures that all alerts are easily traceable and analysed.

The **"AlertManager"** class manages all "Alert" instances sent by the "AlertGenerator".
There are two methods within the class: sendAlert() and addToPatientHistory(). They are needed for alert transmission and creation of historical record. Alerts are managed using a Priority Queue, ensuring prioritising of critical alerts.

Supporting Classes:

The **"PatientRecord"** class maintains detailed patient information, including personalized thresholds ("heartRateThreshold", "bloodPressureThreshold") required for alert analysis.
The class also stores history of all alerts triggered for the patient ("alertHistory"). Collecting all data concerning a patient within one class simplifies the access to the data and performing updates on the data.

The **"PatientData"** class represents real-time patient data. It includes "heartRate" and "bloodPressure" and boolean indicators for whether they fit the established norm. Separation of dynamically changing data from static patient data of "PatientRecord" is important for real-time evaluation performed by the system.

The **"BloodPressure"** class is needed, because of a structure of blood pressure interpretation.

The **"BloodPressureThreshold"** and **"HeartRateThreshold"** support the "PatientRecord" class by defining how the readings of health indicators are performed. Their separation from the more central class "PatientRecord" ensures a distinct declaration and highlights their separateness.