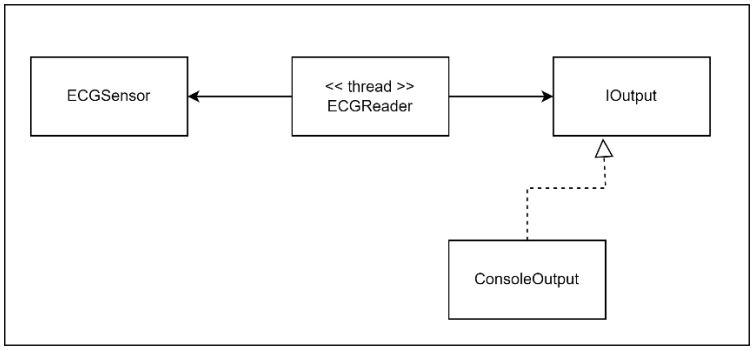
Exercise: Client – Server, Electrocardiography (ECG)

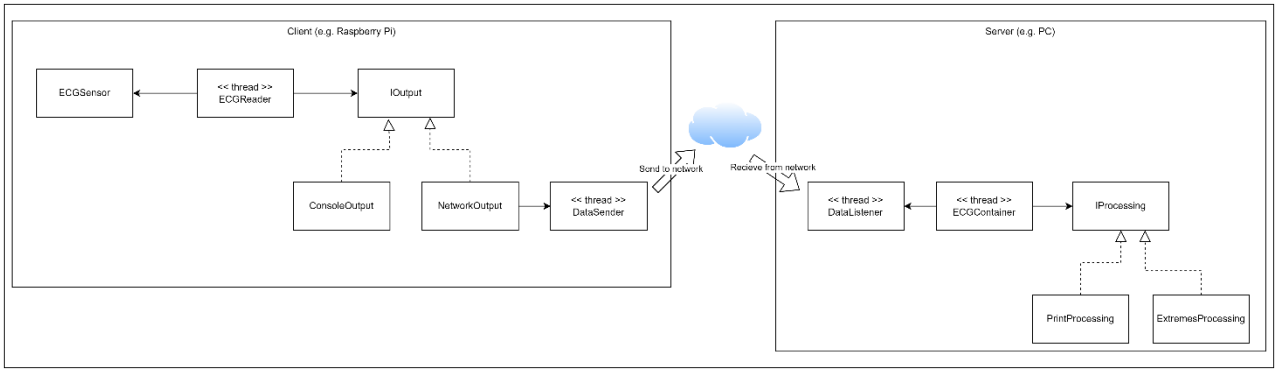
In this exercise, you create two programs:

- One program that reads data from a simulated ECG sensor and sends the data to another program using a TCP connection on an IP network.

In the initial design of the system, the data is printed on the same machine, where it is collected.
The design could look like this:



A design where the producer and consumer are two separate programs will look like this:



Exercise 1:
Implement the initial design, where data is printed on the same machine as it is collected.

The ECGSensor shall have a method to provide a new reading. It shall return a random value between 0 and 100.
Implement the ECGReader class to have a method with an infinite loop, which reads data from the ECGSensor once every second.

The ECGReader shall output the data using the ConsoleOutput class, which implements an IOutput interface.
Create a thread, which runs the ECGReader method with the infinite loop.

Exercise 2:
You shall now modify the program from exercise 1, so you will have the data producer and data consumer in two separate programs.

Create a new program, which will be the server and add a DataListener class to it. It shall listen for incoming data and print the data it receives. For now, you do **not** need to implement the ECGContainer and processing shown in the diagram.

Add a DataSender class to your program. It shall have a method to add new data to be sent.
The DataSender shall run in its own thread and when there is new data to be sent, it shall send the data to the server.

For a start, run both the client and the server on the same machine. Use the loopback address 127.0.0.1 and a port of your own choice.

Exercise 3:
Modify the DataListener class, so it instead of printing the data, just stores the data and has a Boolean property, which indicates when unread data is available.

Create a method to read the data. When called, this method shall return the data and also modify the Boolean property to indicate, that data has been read.

Create an ECGContainer class, with a method in an infinite loop, which reads new data from the DataListener and processes the data by printing it to the console.

Exercise 4:
Make it possible to stop the program, which sends data by pressing 'x' in the console. It should stop without any exceptions in the console.