```
/*
*   fillDataInPackage(String package)
*
*   Args:
*
*       package:    receives the package to fill it with data
*
*   Actions:
*
*       Fills the received package with the sensors data and then return it
*/
String fillDataInPackage(String package)
{

    lastEncoderValue = encoder.getPulseCounter();
    int spinEncoder = (lastEncoderValue - previousEncoderValue);
    previousEncoderValue = lastEncoderValue;

    lastFuelInValue = fluxometerFuelIn.getPulseCounter();
    int fuelIn = (lastFuelInValue - previousFuelInValue);
    previousFuelInValue = lastFuelInValue;

    lastFuelOutValue = fluxometerFuelOut.getPulseCounter();
    int fuelOut = (lastFuelOutValue - previousFuelOutValue);
    previousFuelOutValue = lastFuelOutValue;

    int scaleValue = (int)scale.get_units();

    int frequency = (int)(FreqMeasure.countToFrequency(frequencyReadSum / frequencyReadsCount) * 1000);

    package.concat(MAC);
    package.concat(";");
    package.concat(String(timeToSend, HEX));
    package.concat(";");
    package.concat(String(packageCounter, HEX));
    package.concat(";");
    package.concat(String(spinEncoder, HEX));
    package.concat(";");
    package.concat(String(fuelIn, HEX));
    package.concat(";");
    package.concat(String(fuelOut, HEX));
    package.concat(";");
    package.concat(String(temperatureFuelIn.read(), HEX));
    package.concat(";");
    package.concat(String(temperatureFuelOut.read(), HEX));
    package.concat(";");
    package.concat(String(temperatureEngineAirIn.read(), HEX));
    package.concat(";");
    package.concat(String(temperatureExhaustGases.read(), HEX));
    package.concat(";");
    package.concat(String(temperatureWaterCooling.read(), HEX));
    package.concat(";");
    package.concat(String(engineAirFlow.read(), HEX));
    package.concat(";");
    package.concat(String(scaleValue, HEX));
    package.concat(";");
    package.concat(String(frequency, HEX));

    return package;

}
```