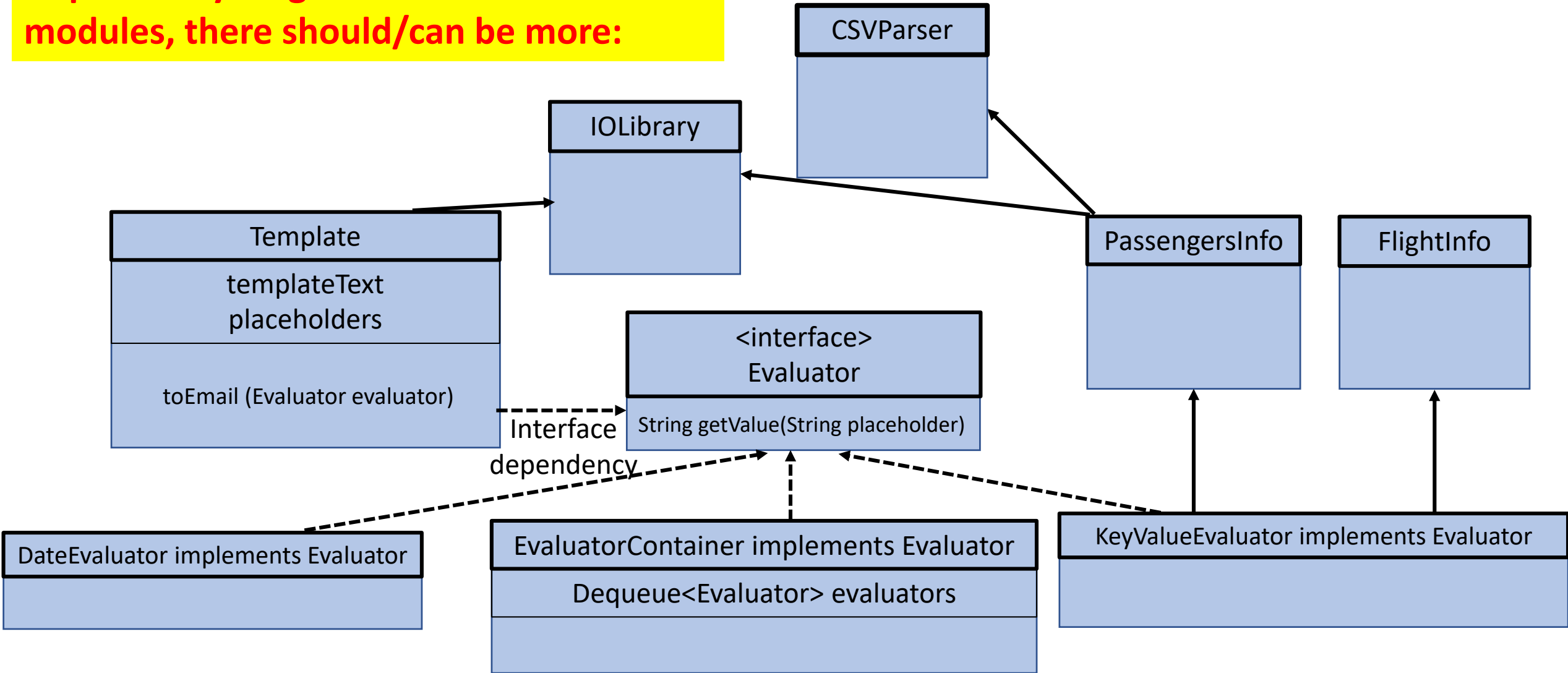# Designing "Top to Bottom" Assignment 3

This is ONLY a suggestion/ one of possible designs

Maria Zontak

Dependency diagram between MAJOR modules, there should/can be more:

CSVParser

IOLibrary

Template

templateText
placeholders

toEmail (Evaluator evaluator)

Interface dependency

<interface>
Evaluator

String getValue(String placeholder)

PassengersInfo

FlightInfo

DateEvaluator implements Evaluator

EvaluatorContainer implements Evaluator

Dequeue<Evaluator> evaluators

KeyValueEvaluator implements Evaluator

```
… //many things like command line arguments handling happened before

Template emailTemplate = new Template(template_filename);
PassengersInfo passInfo = new PassengerInfo(csv_filename);

Evaluator ec = new EvaluatorContainer(); //will contain all sources of info
ec.push(new DateEvaluator());
ec.push(new KeyValueEvaluator(SOURCE_CITY_PLACEHOLDER, FlightInfo.getSourceCity()));
ec.push(new KeyValueEvaluator(DESTIN_CITY_PLACEHOLDER, FlightInfo.getDestinationCity()));
ec.push(new KeyValueEvaluator(EVENT_PLACEHOLDER, FlightInfo.getEvent()));

List<String> headers = passInfo.getHeaders();
while( passInfo.hasNextPassenger()) {
      ec.push(new KeyValueEvaluator(headers, passInfo.nextPassenger());
      Email newEmail = emailTemplate.toEmail(ec);
      ec.pop(); //remove evaluator that was added last
      // send/save your newEmail
      ...
}
```