School of Computing and Information Technology
University of Wollongong

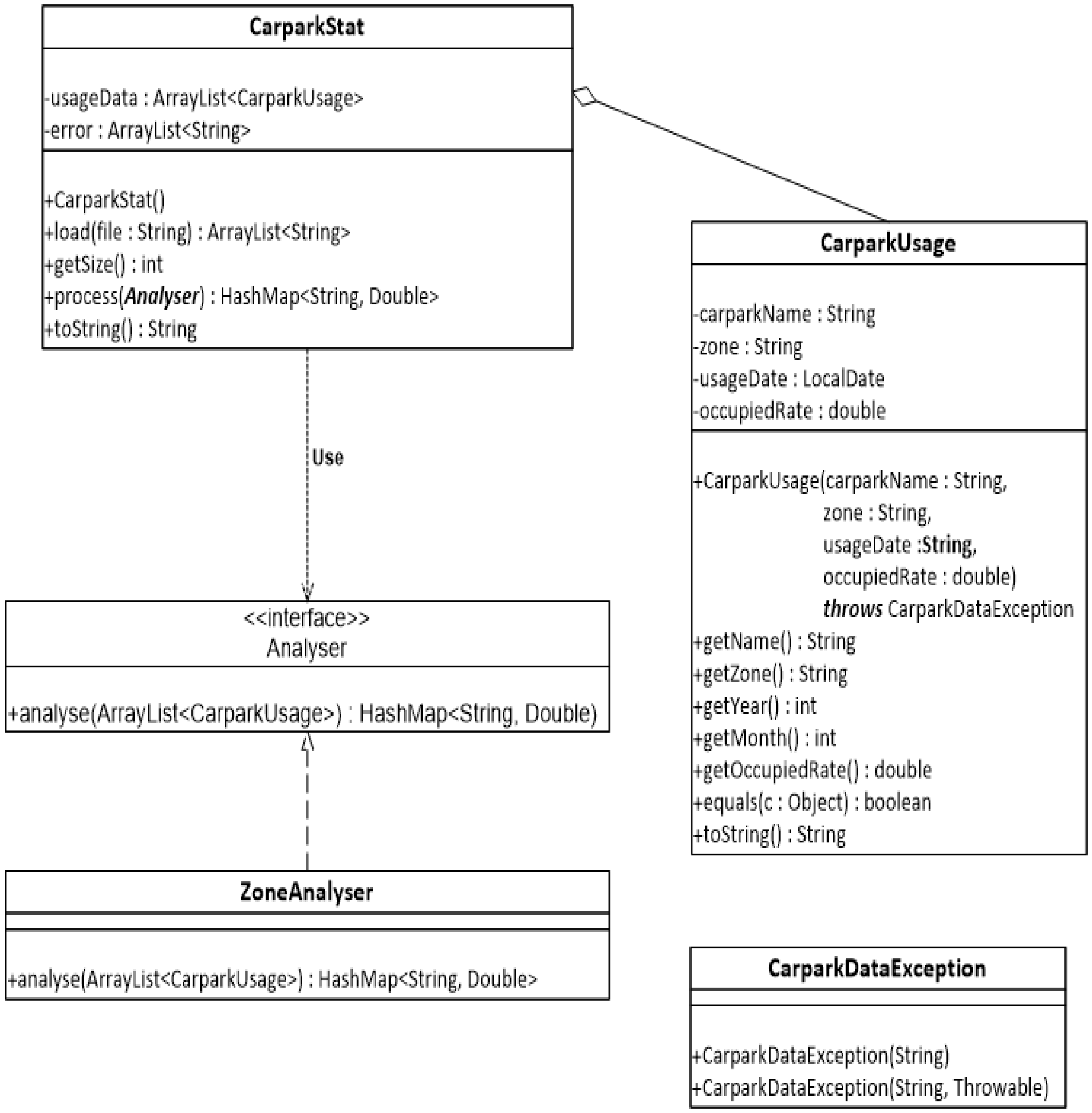
CSIT213 Java Programming
Assignment 3 (15%)
Due: refer to Moodle

Objectives

- To apply Object Oriented Design.
- To apply Object Oriented Programming.
- To implement solution in Java and IDE.

Tasks

The following class diagram is the prototype of an application which stores and analyses carpark usage in the various zones. You are to implement the interface and classes dictated in the class diagram, as well as an additional class containing the main method that tests the classes.


CarparkStat
-usageData : ArrayList<CarparkUsage>
-error : ArrayList<String>
+CarparkStat()
+load(file : String) : ArrayList<String>
+getSize() : int
+process(Analyser) : HashMap<String, Double>
+toString() : String
Use
<<interface>>
Analyser
+analyse(ArrayList<CarparkUsage>) : HashMap<String, Double)
ZoneAnalyser
+analyse(ArrayList<CarparkUsage>) : HashMap<String, Double>
CarparkUsage
-carparkName : String
-zone : String
-usageDate : LocalDate
-occupiedRate : double
+CarparkUsage(carparkName : String, zone : String, usageDate :String, occupiedRate : double) throws CarparkDataException
+getName() : String
+getZone() : String
+getYear() : int
+getMonth() : int
+getOccupiedRate() : double
+equals(c : Object) : boolean
+toString() : String
CarparkDataException
+CarparkDataException(String)
+CarparkDataException(String, Throwable)

CarparkUsage

An instance of this class is a record of the usage of a carpark on a particular day.

carparkName	Unique carpark name.
zone	Each carpark belongs to a zone.
usageDate	Date of the usage record.
occupiedRate	Percentage of occupied lots on a particular date (usageDate).
CarparkUsage	The constructor receives parameters to initialise **all** the instances variables. The constructor throws ***CarparkDataException*** with an appropriate message for the following errors: • usageDate does not represent a valid date • occupiedRate does not fall in the range of 0 ~ 100 (inclusive at both ends)
getName	Returns carparkName.
getZone	Return zone.
getYear	Returns the year of the usageDate.
getMonth	Returns the month of the usageDate.
getOccupiedRate	Returns occupiedRate.
equals	Two instances are considered equal if they have the same carparkName **and** usageDate.
toString	Returns a String containing all the instance variables.

CarparkStat

This class provides methods to load and analyse the carpark usage records.

usageData	ArrayList of CarparkUsage instances.
error	ArrayList of String instances representing erroneous records detected in the data file.
CarparkStat	No parameter.
load	Loads the records from a data file into UsageData. Returns an ArrayList<String> containing erroneous data found in the data file. If there is no error, returns an empty ArrayList<String>.

getSize	Returns the number of instances in usageData.
process	Uses an instance of ***Analyser*** to analyse the records. Returns a HashMap<String, Double> containing the analysis result.
toString	Returns a String containing all the instances in UsageData.

ZoneAnalyser

This class implements ***Analyser***. It calculate the average occupied rate of the carparks in each zone.

analyse	Returns a HashMap<String, Double> such that for each entry in the HashMap: • key (String) is the zone name • value (Double) is the ***average*** occupiedRate of all the carparks in the same zone.

Data file

Each line in the text file represents an instance of CarparkUsage. Below is an example of a date file.

```
2025-02-10,cp 1,zone 1,10
2025-02-10,cp 2,zone 1,20
2025-02-10,cp 3,zone 1,30
2025-02-10,cp 4,zone 1,40
2025-02-10,cp 5,zone 1,50
2025-15-32,cp X,zone X,50
2025-02-11,cp 21,zone 2,20
2025-02-11,cp 22,zone 2,40
2025-02-11,cp 23,zone 2,5
2025-02-11,cp 24,zone 2,35
2025-02-12,cp 1,zone 1,30
2025-02-12,cp 2,zone 1,20
2025-02-12,cp 3,zone 1,40
2025-02-12,cp 4,zone 1,50
2025-02-13,cp 21,zone 2,10
2025-02-13,cp 22,zone 2,20
2025-02-13,cp 23,zone 2,30
2025-02-13,cp 24,zone 2,40
2025-02-10,cp 5,zone 1,40
2025-02-14,cp 31,zone 3,31
2025-02-10,cp X,zone 1,40, 100
2025-02-14,cp 32,zone 3,32
2025-02-10,cp Y,zone 1
2025-02-14,cp 33,zone 3,33
2025-02-14,cp 1,zone X, -5
2025-02-14,cp 34,zone 3,24
```

- You may assume that values in each line are ordered correctly::
    - usage date, carpark name, zone, occupied rate.
- You may assume that the carpark name and zone are correct.
- You may assume that the occupied rate are numeric.
- You must **not** assume that the usage date is valid.
- You must **not** assume that the occupied rate are valid.
    - Valid value is between 0 ~ 100, inclusive at both ends.
- You must **not** assume that the number of values per line is valid.
- You must **not** add invalid data into the ArrayList.
- You must **not** add duplicate instances into the ArrayList.

Requirements

- Please include the following information at the beginning of the Java source file.
    - Module code
    - Assignment name
    - Your UOW student number.
    - Your full name.
    - Your tutorial group.
- You must adopt good programming practices which include (but not limited to) the followings:
    - Descriptive variable, parameter and method names.
    - Readable source code with proper indentation.
    - Descriptive block comments in the source code and the test results.
    - Modular programming.
    - Avoid global variables.
    - Exception handling
- You must include sufficient test cases to thoroughly test the classes and the methods implemented in your solution.
- You should organize the sequence of the test cases so that the test results are readable.

Submission

- Please submit ***one zip file*** containing the following (do not place java class in a package):
    - CarparkUsage.java
    - CarparkStat.java
    - Analyser.java
    - ZoneAnalyser.java
    - CarparkDataException.java
    - StudentAssertionTest.java
    - output.txt (execution result)
- Only zip file is accepted. Please do not use other compression format such as 7x, grip, rar and etc.
- For full time student, the zip file name must be in the form of: TXX**F**_NAME_UOWID.zip where XX is your tutorial group, NAME is your full name (without space and underscore) and UOW is your 7-digit UOW ID (**not** SIM ID).
- For part time student, the zip file name must be in the form of: TXX**P**_NAME_UOWID.zip where XX is your tutorial group, NAME is your full name (without space and underscore) and UOWID is your 7-digit UOW ID (**not** SIM ID).

- Please ensure the UOWID is correct as it will be used by the marking scripts to update your assignment mark in Moodle. Incorrect ID will result in 0 mark in your Moodle.
- Automated scripts will be used to extract your files for grading. Failure to follow the above instructions will result in failure to extract and grade your submission, and 0 mark will be awarded.
- No appeal will be granted for incorrect submission such as empty zip file, wrong files, etc.
- Late submission will be penalized 5% per day late.
- Late submission after 4 days will not be graded.