Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 03 REPORT

UĞURKAN ATEŞ
151044012

Course Assistant:M.BURAK KOCA

1 INTRODUCTION

1.1 Problem Definition

Part 1

We need a data structure that able to keep all GTU Computer Engineering courses as nodes to manipulate courses and write methods to do so. As seen from homework instructions extending of already in place systems like Java Linked List is big no-no so we are going to need a new solution for this matter.With this new structure user of this data type should able to return courses with given criteria(such as course codes,choice of semester or a range within indexes) to a linked list. And do as she/he pleases with returned list such as printing course values or data manipulation for later uses.

Part 2

We need a extended version of Java Linked List structure to do add new functionality such as enable , disable and able to show all disable nodes in linked list.A disable method should be removed from list so it shouldnt able to use methods like get,set,remove or any other methods of other nodes.

Part 3

We need a new data structure to still keep courses as linked list before such in part 1-2 but also link same type semester courses linked with eachother. To able to that we had to come with new structure that keeps some elements of Linked List structure but also new methods – and data types to link semester with eachother.

1.2 System Requirements

Part 1

I needed a object instance of Java's linked list structure that is already implemented. So i was able to use that instance later on for new methods i wrote. I needed a course class to keep data of each course and i needed this class to be future viable. To keep continue using same course class structure later on for part 2 and part 3. I needed a CSV Reader class for all data manipulations , as seen from it we are reading data from .csv file containing class all seperated with ",". I had to made this class future viable too because all other classes uses same kind of data to use with.This course structure had all parts of GTU computer science courses as data fields to manipulate or check on later.Such as part

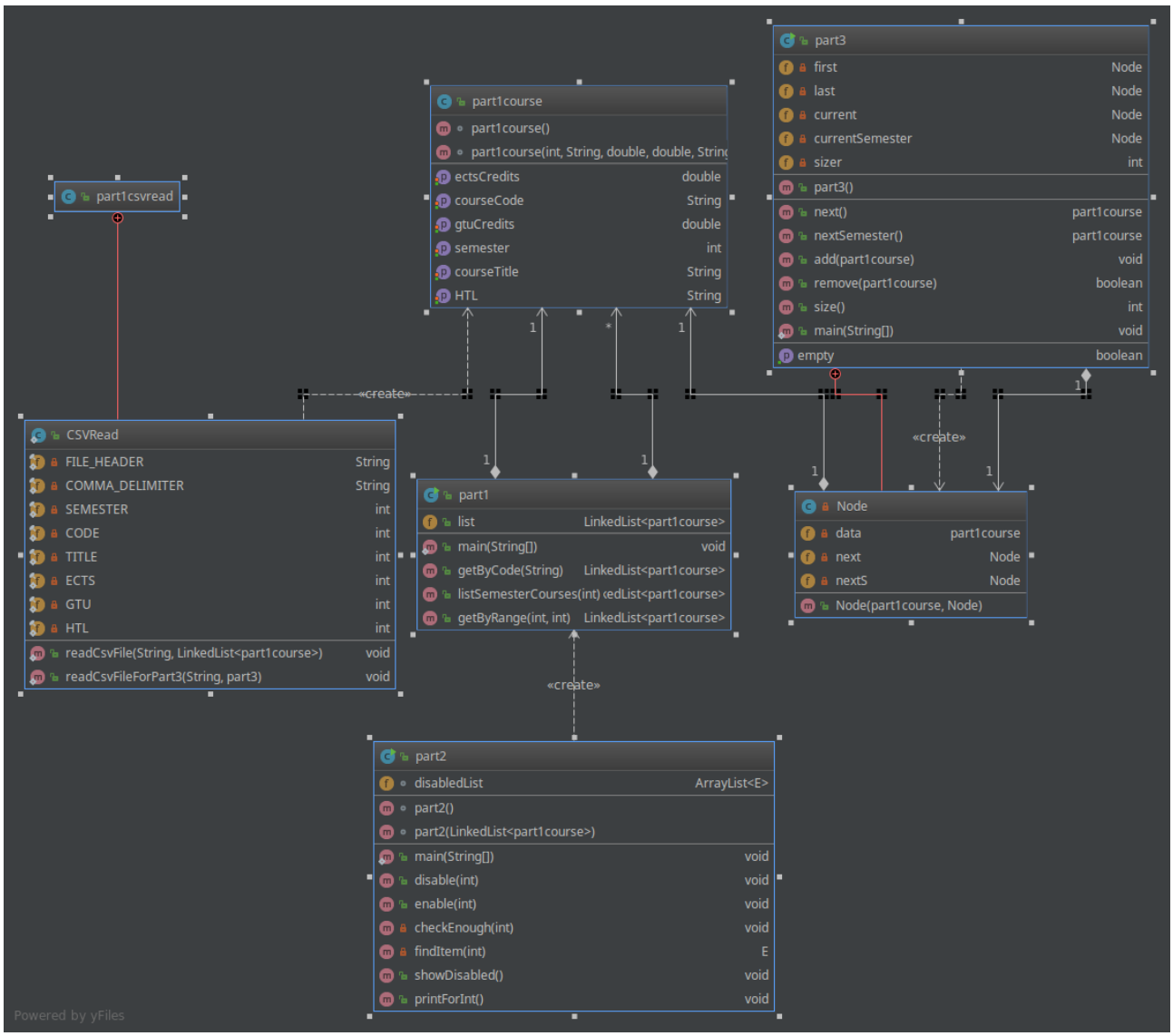
1 needed “getByCode” method which returns linked list of given course code. So i had to keep a course code data field for later manipulations. I extracted all data from CSV with my CSV reader. And implemented getByCode,listSemesterCourses,getByRange methods accordingly. For this methods I used linked list’s iterator mostly but for getByRange method i used for loop with given parameters as indexes.

Part 2
I extended Java’s Linked List implementation to give ability enable/disable to class. To do that i had to keep indexes and removed(disabled) items in seperate list so those elements and their indiviual indexes wouldnt lost but they wouldnt be able to use normal link list methods such as set,get etc..I needed a seperate list for disabledItems.

Part 3
I had to make with my own courseStructure that is competable with my coursesClass implemented in part 1. This data structure albeit a similiar to linked list official implementation it had clear differences such as all same semester class were linked together.To do that i needed another kind of Iterator other than current to keep nextSemester always.

2 METHOD

2.1 Class Diagrams



2.2 Use Case Diagrams

-> Add use case diagrams if required

Didnt think they were required for this homework.

2.3 Problem Solution Approach

Part 1

I declared an object of LinkedList and used instance of that list in my methods.For getByCode and listSemesterCourses methods i simply iterated iterator of object instance and checked if it fits given parameter from method. If it did i added to list. I returned object instances i created within methods. For ranged method instead of iterating methods i used for loop within size of parameters wanted. I double checked if list didnt go higher than it holds. (such as giving 5 to 3 element linked list). And throw exceptions if necessary.

CSV reader was mostly same from Homework01 with current course structure.

Part 2

To solve some of problem definitions i already complained , i hold an ArrayList for disabled objects. And made that ArrayList such as being able to keep elements of given disableElement. If user wanted 8th element disabled in main list i made sure there was 8 element in array list, if not i made it. Reason for this decision and not being able to return a class with index instead(such as 8th element goes 0th element in another list) homework expliclity wanted this class being able to work with any E generic class. This only made possible with this solution. Because if i used another class in the end when enabling same method it Java Compiler had to turn course object to disableClass object instance. While i deep looked into the problem then i saw this is one of the short comings of Java's Generic class problem such as there are limits of what generics could do or not. If this linked list class was only viable for course class the other implementation would fit perfectly and honestly a better solution to problem.

Part 3

I basically implemented a basic linked list with(single linked list) implementation and hold use of 4th iterator for me to keep track of NextSemester method work. First,Last and current nodes are all normal structure variables but i also added currentS to keep track of nextSemester.Such as when nextSemester() called currentS assigned to current node and when i found nextSemester i simply made current also what currentS is pointing. This way it works perfectly without any performance hitch ups.Aside from these i implemented all methods wanted from homework such as add,remove,next,nextS,size. Those are all classic link list methods nothing unusual or different from these methods.

Part 4 Bonus – Performances

Part 1->Perf

main:

object creation: constant time

csvReader: O(n) time

getByCode: O(n) time

listSemester : O(n) time

geyByRange : O(n) time

in final main : O(n) time

Part 2 -> Perf

main:

object creation : constant time

2nd object creation : constant time

csvReader : O(n) time

disableMethod->

checkEnough: O(n) time

add – remove constant time

O(n) time

enable : constant time(add to last remove from last)

in final : O(n) time

Part 3 -> Perf

object creation: constant time

csvReader: ?? unknown

next() constant time

printer : constant time

nextSemester() -> unknown-> worstCase O(N) -> so O(n)

final: O(n) time

3 RESULT

3.1 Test Cases

Part 1

for getByCode I created a part 1 object.Read whole of csv with my csvreader class to list in part1 object.After that tried and sent “CSE312” parameter to method and expected to return 2 element linked list. (I added CSE312 class twice to get 2 element linked list). Then printed list courses on screen with my printer method.

for List SemesterCourses i created part 1 object.Read CSV to object.And wanted to list SEMESTER 1 courses. It returned instance of all courses in semester 1. I printed out that linked list and it shows correctly.

For getByRange I created object and read csv to object. Given parameters 20 to 30 it returned a linked list containg csv courses 21-31 considering first is just header.

Part2

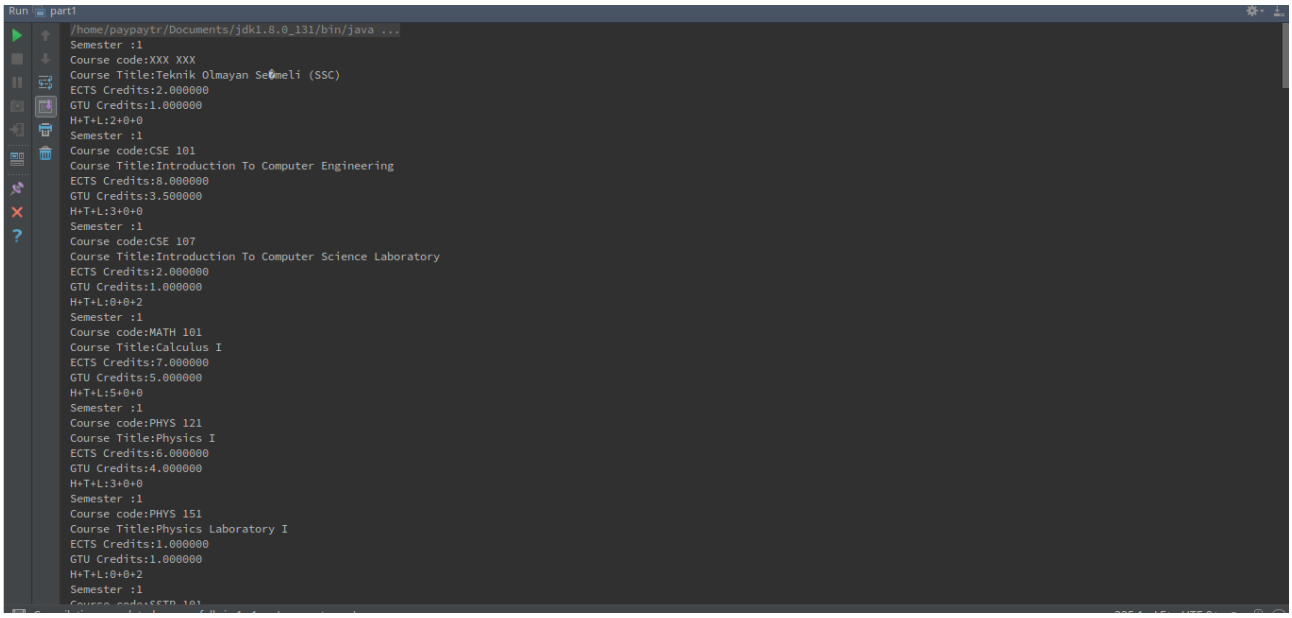
for disableMade object and initted with csv read list. I disabled 18th element and showed on disabled list.

For enable made object and initted with csv read list. Disabled 19th element and showed on screen, then re enabled again and showed disable list again it shows no element.

Part 3

For unit tests i made some random courses and tested with them. For main test in part3.java I read entire csv and iterated through first semester with nextSemester() and printed each course on screen as move along. In the end when last of first semester came, it went back to first of first semester courses.

3.2 Running Results



```
This 18 numbered item from list is disabled.Re-enable to use methods on this item.

Process finished with exit code 0
```

```
Course code:XXX XXX
SEMESTER:1
ECTS and GTU:2.0000001.000000
Course name :Teknik Olmayan Se𝍐meli (SSC)
Course code:CSE 107
SEMESTER:1
ECTS and GTU:2.0000001.000000
Course name :Introduction To Computer Science Laboratory
Course code:MATH 101
SEMESTER:1
ECTS and GTU:7.0000005.000000
Course name :Calculus I
Course code:PHYS 121
SEMESTER:1
ECTS and GTU:6.0000004.000000
Course name :Physics I
Course code:PHYS 151
SEMESTER:1
ECTS and GTU:1.0000001.000000
Course name :Physics Laboratory I
Course code:SSTR 101
SEMESTER:1
ECTS and GTU:2.0000002.000000
Course name :Principles Of Atat𝍐rk And The History Of Turkish Revolution I
Course code:TUR 101
SEMESTER:1
ECTS and GTU:2.0000002.000000
Course name :Turkish I
Course code:XXX XXX
SEMESTER:1
ECTS and GTU:2.0000001.000000
Course name :Teknik Olmayan Se𝍐meli (SSC)

Process finished with exit code 0
```