CS5004 Lab06 Report

Rong Huang

Lab 6: Abstracting Out a Tree

1. Reflection

- **Understanding Hierarchies:** I learned how hierarchies can be represented in code. Through the abstraction of a tree structure, I learned how to model organizational relationships, where each node can represent an employee, and the connections between them depict managerial relationships.
- **Code Refactoring with Hierarchies:** Another significant learning point was how to refactor existing code to utilize a hierarchical model. This involved rethinking the way data was structured and accessed, moving from perhaps a more linear or flat structure to a multi-level tree structure that more naturally mirrors real-world organizational charts.
- **Functional Programming Techniques**: Working with Java's functional programming interfaces like Predicate, Function, and BiFunction within the context of a tree structure deepened my understanding of how these interfaces can be utilized to manipulate and process data in complex structures.

2. Extensions

1) extra task 1: print the task without “the”

implement the List<String> allEmployees(Predicate<Employee> predicate) in OrganizationImpl.java

```java
@Override
public List<String> allEmployees(Predicate<Employee> predicate){
    List<Employee> employees = root.toList();
    List<String> result = new ArrayList<>();
    for (Employee employee : employees) {
        if (predicate.test(employee)) {
            result.add(employee.getName());
        }
    }
    return result;
}
```

```java
//Extra Task
//Because we aren't implementing a filter, I'm removing this, but I invite you to find a workaround
System.out.println(MonsterCorp.allEmployees(m -> !(m.getName().contains("the"))));
```

2) extra task 3: print the number of employees terminated before specified date
implement the public int terminatedBefore(int date, int month, int year) in OrganizationImpl.java

```java
@Override
public int terminatedBefore(int date, int month, int year) {
    LocalDate threshold = LocalDate.of(year, month, date);
    // Maps employees whose termination date is before the specified threshold to 1, others to 0
    // then sums these values to count employees meeting the condition.
    return root.map(e -> {
        if (!e.getEmploymentEndDate().equals("XXXXXXXX")) {
            LocalDate endDate = LocalDate.parse(e.getEmploymentEndDate(), DateTimeFormatter.ofPattern("MMddyyyy"));
            return endDate.isBefore(threshold) ? 1 : 0;
        }
        return 0;
    }).reduce( initialValue: 0, (a, b) -> a + b);
}
```

```java
//Extra Task
//Implement some new functionality on your own or add back some of the functionality that we had to remove
// Example usage of terminatedBefore
int countTerminatedBefore = MonsterCorp.terminatedBefore( date: 1,  month: 1,  year: 2024);
System.out.println("Number of employees terminated before specified date: " + countTerminatedBefore);
```

3. Grading Statement

Add the extensions, total 100.

4. Academic Integrity Statement

I understand that my learning is dependent on individual effort and struggle, and I acknowledge that this assignment is a 100% original work and that I received no other assistance other than what is listed here.

Acknowledgements and assistance received:
Professor Molly, TA Will
Google, StackOverFlow

I did not use generative AI in any form to create this content and the final content was not adapted from generative AI created content.

I did not view content from any one else's submission including submissions from previous semesters nor am I submitting someone else's previous work in part or in whole.

I am the only creator for this content. All sections are my work and no one else's with the exception being any starter content provided by the instructor. If asked to explain any part of this content, I will be able to.

By putting your name and date here you acknowledge that all of the above is true and you acknowledge that lying on this form is a violation of academic integrity and will result in no credit on this assignment and possible further repercussions as determined by the Khoury Academic Integrity Committee.

Name:Rong Huang	Date: 04/03/2024