CS5004 Lab02 Report

Rong Huang

Lab 02: Employees and Paychecks

1. Reflection(*What did you learn?*)

1) Understanding Class Design and Static Variables:

The use of static variables in the Employee class, specifically for tracking the number of Employee instances, was a key learning aspect. It demonstrated how static variables can be effectively employed to maintain class-level data, which is crucial in scenarios where shared information needs to be managed across all instances. This concept was particularly useful in understanding the broader aspects of class design in object-oriented programming.

2) Implementing and Handling Exceptions:

The implementation of exception handling in the project was an essential exercise in writing resilient software. It highlighted the importance of considering and planning for exceptional cases, such as null values and invalid inputs. This aspect of the project underscored the necessity for developers to anticipate potential issues and incorporate safeguards to ensure the stability and reliability of the software.

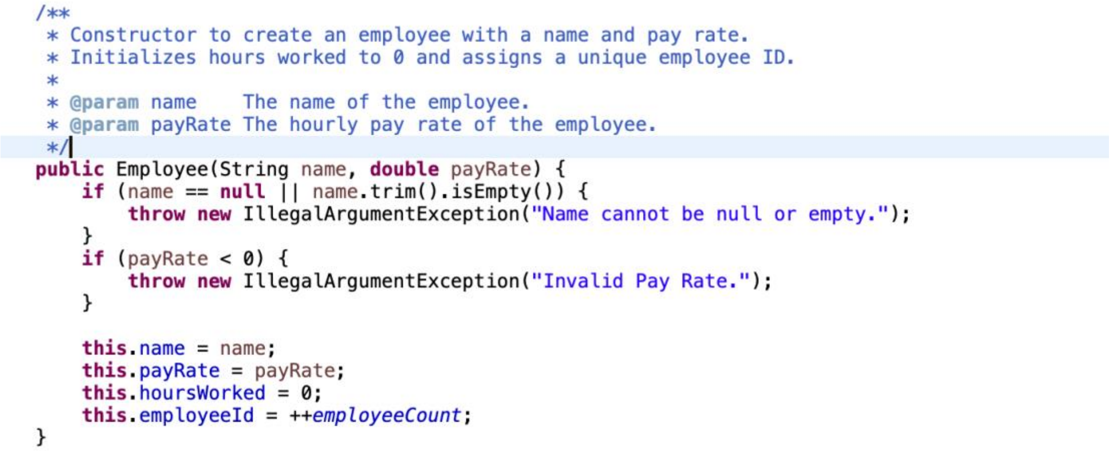
3) Exploring Enums and Advanced Features

Integrating Enums to manage payment methods in the Employee and Paycheck classes was an insightful extension of the project. It not only provided a practical application of Enums but also added a layer of complexity and realism to the software. This exploration into Enums and their utility in a real-world context was an enriching experience, furthering my understanding of advanced Java features and their practical applications in software development.

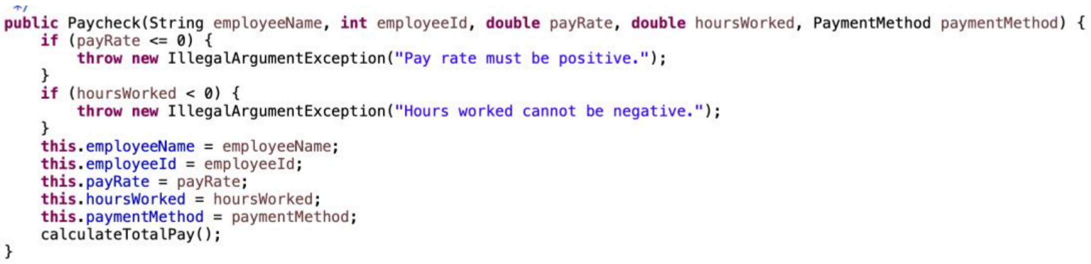
2. Extensions (What extensions are you requesting?)

1) Add in Exception handling in at least three more places, outside of what is requested for the driver

I add extra Exception handling in Employee constructor handling the IllegalArgumentException with name. And also add extra Exception handling in paycheck constructor handling the IllegalArgumentException with payrate and hoursWorked.



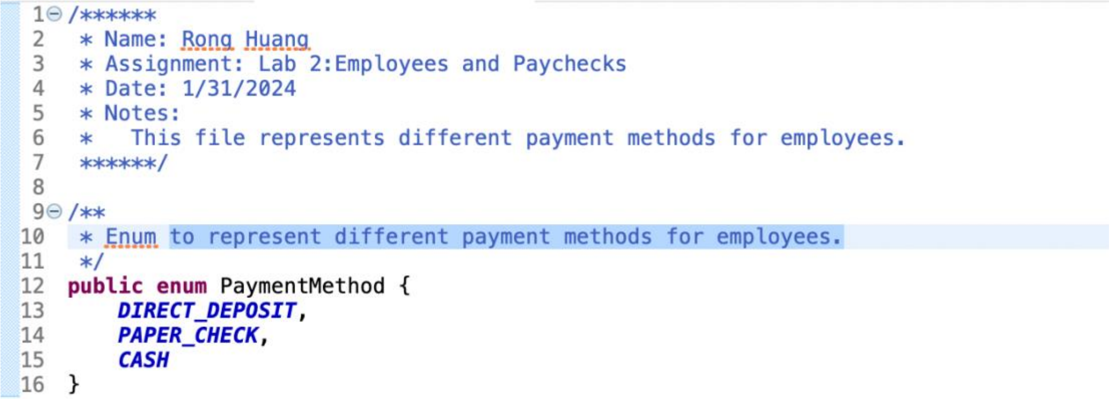
```java
/**
 * Constructor to create an employee with a name and pay rate.
 * Initializes hours worked to 0 and assigns a unique employee ID.
 *
 * @param name    The name of the employee.
 * @param payRate The hourly pay rate of the employee.
 */
public Employee(String name, double payRate) {
    if (name == null || name.trim().isEmpty()) {
        throw new IllegalArgumentException("Name cannot be null or empty.");
    }
    if (payRate < 0) {
        throw new IllegalArgumentException("Invalid Pay Rate.");
    }

    this.name = name;
    this.payRate = payRate;
    this.hoursWorked = 0;
    this.employeeId = ++employeeCount;
}
```



```java
 */
public Paycheck(String employeeName, int employeeId, double payRate, double hoursWorked, PaymentMethod paymentMethod) {
    if (payRate <= 0) {
        throw new IllegalArgumentException("Pay rate must be positive.");
    }
    if (hoursWorked < 0) {
        throw new IllegalArgumentException("Hours worked cannot be negative.");
    }
    this.employeeName = employeeName;
    this.employeeId = employeeId;
    this.payRate = payRate;
    this.hoursWorked = hoursWorked;
    this.paymentMethod = paymentMethod;
    calculateTotalPay();
}
```

2) Creat an Enum and using it within my code

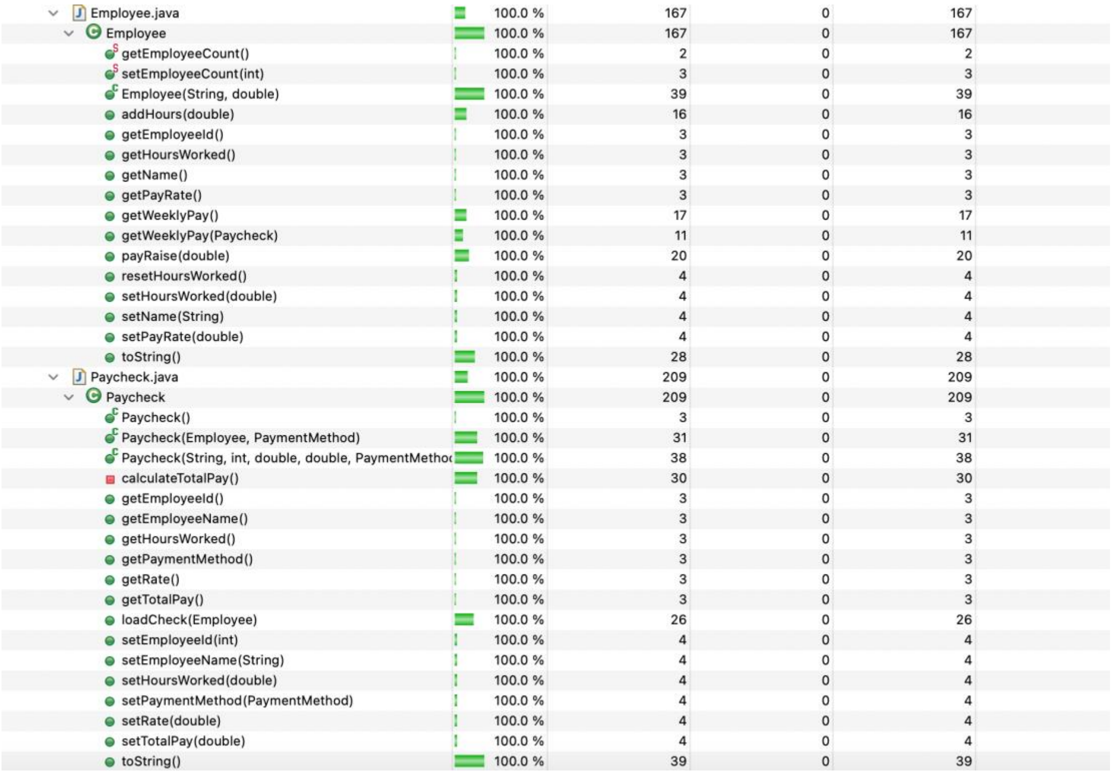
I created an Enum to represent different payment methods for employees, using it in my code and finally print the payment method in my driver.



```java
/******
 * Name: Rong Huang
 * Assignment: Lab 2:Employees and Paychecks
 * Date: 1/31/2024
 * Notes:
 *   This file represents different payment methods for employees.
 ******/

/**
 * Enum to represent different payment methods for employees.
 */
public enum PaymentMethod {
    DIRECT_DEPOSIT,
    PAPER_CHECK,
    CASH
}
```

3) Reach the highest possible coverage for testing

The Employee’s and Paycheck’s test covered 100%, which reaching the highest possible coverage for testing.



Employee.java	100.0 %	167	0	167
Employee	100.0 %	167	0	167
getEmployeeCount()	100.0 %	2	0	2
setEmployeeCount(int)	100.0 %	3	0	3
Employee(String, double)	100.0 %	39	0	39
addHours(double)	100.0 %	16	0	16
getEmployeeId()	100.0 %	3	0	3
getHoursWorked()	100.0 %	3	0	3
getName()	100.0 %	3	0	3
getPayRate()	100.0 %	3	0	3
getWeeklyPay()	100.0 %	17	0	17
getWeeklyPay(Paycheck)	100.0 %	11	0	11
payRaise(double)	100.0 %	20	0	20
resetHoursWorked()	100.0 %	4	0	4
setHoursWorked(double)	100.0 %	4	0	4
setName(String)	100.0 %	4	0	4
setPayRate(double)	100.0 %	4	0	4
toString()	100.0 %	28	0	28
Paycheck.java	100.0 %	209	0	209
Paycheck	100.0 %	209	0	209
Paycheck()	100.0 %	3	0	3
Paycheck(Employee, PaymentMethod)	100.0 %	31	0	31
Paycheck(String, int, double, double, PaymentMethoc	100.0 %	38	0	38
calculateTotalPay()	100.0 %	30	0	30
getEmployeeId()	100.0 %	3	0	3
getEmployeeName()	100.0 %	3	0	3
getHoursWorked()	100.0 %	3	0	3
getPaymentMethod()	100.0 %	3	0	3
getRate()	100.0 %	3	0	3
getTotalPay()	100.0 %	3	0	3
loadCheck(Employee)	100.0 %	26	0	26
setEmployeeId(int)	100.0 %	4	0	4
setEmployeeName(String)	100.0 %	4	0	4
setHoursWorked(double)	100.0 %	4	0	4
setPaymentMethod(PaymentMethod)	100.0 %	4	0	4
setRate(double)	100.0 %	4	0	4
setTotalPay(double)	100.0 %	4	0	4
toString()	100.0 %	39	0	39

3. Grading Statement (Based on the rubric, what grade do you feel your work should be? Be honest, don't be humble, and use the rubric.)

Employee 5*7=35
Paycheck 5*4=20
Junit Tests 5*5=25
Misc 5
Extensions 10
Creative or went above and beyond 10
Total 105

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:
TA Will
Canvas moduel 2
Java Enums Explained in 6 Minutes:
https://www.youtube.com/watch?v=wq9SJb8VeyM
#3 Junit Tutorial - Junit Annotations | @Before | @After | @BeforeClass @AfterClass: https://www.youtube.com/watch?v=oAqgFpzv2LY

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:1/31/2024