
MINNESOTA INCOME TAX CALCULATION- REFACTORING

OVERALL REPORT

VERSION <1.0>

BOZ NTOURAN 2310

TABLE OF CONTENTS

Introduction	4
Refactored Design	4
Use Cases	4
Architecture	5
Detailed Design	5
Classes Responsibilities and Collaborations (CRC CARDS)	5

INTRODUCTION

The main objective of this project is to refactor the old code so it can be more easily and faster maintained and extend than its previous version. Another objective is to change the UI to a more user friendly UI.

REFACTORED DESIGN

USE CASES

Use case ID	<u>1</u>
Actors	User
Pre conditions	Files of the correct format existing in the user computer.
Main flow of events	<u>1- Opening the application</u> 2-(Optional) Selecting a file format XML or txt. The default format is txt. 3-User can open a profile either by writing a tax payer number or by loading all the available tax payer on the computer.
Post conditions	The user is ready to do the other use cases described below.

Use case ID	<u>2</u>
Actors	User
Pre conditions	The actor must have loaded a tax payers profile.
Main flow of events	1-)The actor must select one of the loaded profile so he is able to view the tax payers information.
Post conditions	<u>The ability to see the loaded information.</u>

Use case ID	<u>3</u>
Actors	User
Pre conditions	The actors has loaded atleast one profile and has select one of them.
Main flow of events	<p>1-) Click on the receipt tab.</p> <p>2-)Select one of the available receipt.</p> <p>3-)Click on the receipt menu.</p> <p>3.1-)Click View selected receipt.</p>
Alternative flow 1	<p><u>1-)Click on the receipt tab.</u></p> <p>2-)Select a receipt.</p> <p>3-)At the tab below click on the receipt and all information of the receipt is available.</p>
Post conditions	The ability to view and edit a receipt.

Use case ID	<u>4</u>
Actors	User
Pre conditions	Files of the correct format existing in the user computer.The user has selected a receipt.
Main flow of events	1-)Selecting a receipt 2-)Change details of the receipt. 3-)Click on the receipt menu. 3.1-)Select save change.
Post conditions	<u>The actor has successfully modified a saved receipt.</u>

Use case ID	<u>5</u>
Actors	User
Pre conditions	Files of the correct format existing in the user computer. The actor has loaded a tax payer profile.
Main flow of events	<p>1-)Click on the receipts menu</p> <p>1.1-)Click on the add new receipt.</p> <p>1.2-)At the new window fill the details of the receipt.</p> <p>1.3-)Press save.</p>
Post conditions	<u>The user has successfully added a new receipt.</u>

Use case ID	<u>6</u>
Actors	User
Pre conditions	Files of the correct format existing in the user computer.The actor has selected a receipt.
Main flow of events	1-)Select receipt 2-)Click on the receipt menu 2.1-)Click on the delete selected receipt.
Post conditions	The actor has successfully deleted the selected receipt.

Use case ID	<u>7</u>
Actors	User
Pre conditions	Files of the correct format existing in the user computer.The actor has loaded and selected a tax payer profile.
Main flow of events	1-) Select one profile. 2-)Click on the tax payers menu. 3-)Click on delete selected ta payer
Post conditions	The actor will have sucefully deleted a ta payers informations.

ARCHITECTURE

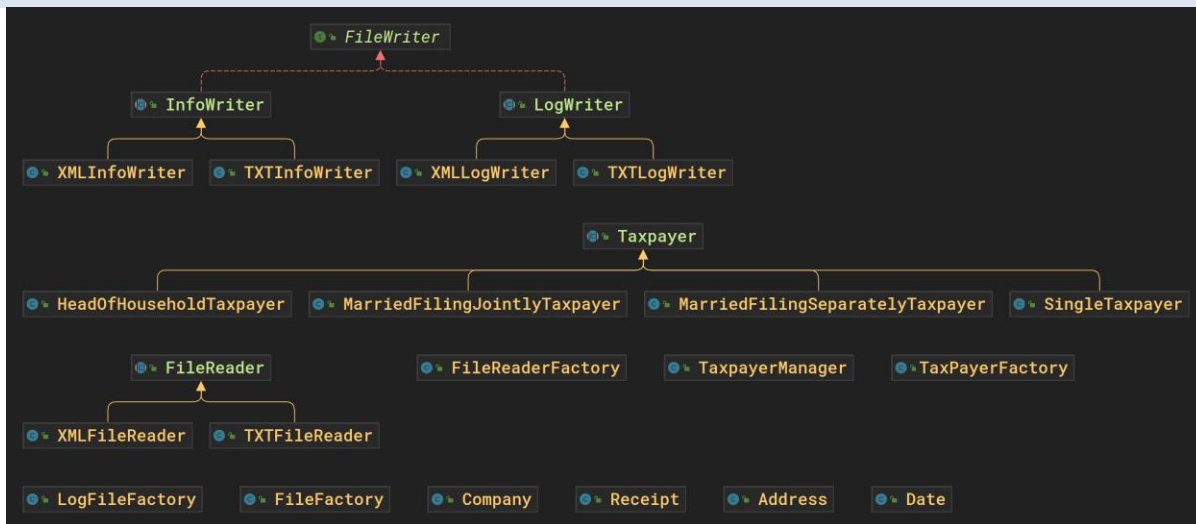
`package incometaxcalculator.exceptions`


`package incometaxcalculator.data`

`package incometaxcalculator.gui`


-

DETAILED DESIGN




 ReceiptAlreadyExistsException

 WrongTaxpayerStatusException


 WrongReceiptDateException


 WrongReceiptKindException

 WrongFileEndingException

 WrongFileFormatException

■

 **GraphicalInterface**

 **NewReceiptDialog**

 **ChartDisplay**



CLASSES RESPONSIBILITIES AND COLLABORATIONS (CRC CARDS)

Class Name:FileFactory	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class is responisble to create the diferent file types the application supports.</u> 	

Class Name:FileReader	
Responsibilities	Collaborations
<p><u>This abstract class is using the template patter to read the different file type the application supports.</u></p>	<ul style="list-style-type: none"> ▪ <u>XMLFileReader nad TXTFileReader implements this class.</u>

Class Name:FileReaderFactory	
Responsibilities	Collaborations
<u>Factory pattern is implemented with this class for reading the supported file types.</u>	<ul style="list-style-type: none"> ▪ <u>List the dependencies and associations of this class with the other classes of the project</u>

Class Name:FileWriter	
Responsibilities	Collaborations
<u>Interface implemented for file wrtitors</u>	<u>This interface is implemented by LogWriter and InfoWriter.</u>
Class Name:InfoWriter	

Responsibilities	Collaborations
<u>This class is implementing the template design pattern for the different file types.</u>	<u>This is implemented by XMLInfoWriter and TXTInfoWriter.</u>

Class Name:LogFileFactory	
Responsibilities	Collaborations
<u>Factory design pattern for log files.</u>	<ul style="list-style-type: none"> ▪ <u>List the dependencies and associations of this class with the other classes of the project</u>

Class Name:LogWriter	
Responsibilities	Collaborations
<u>Abstract class for log file</u>	<u>Implements FileWriter interface and is begin implement by XMLLogWriter and TXTLogWriter.</u>
Class Name:TXTFileReader	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>Read files of specific type txt.</u> 	<ul style="list-style-type: none"> ▪ <u>Extends FileReader</u>

Class Name:TXTInfoWriter	
Responsibilities	Collaborations
<u>Write the Info file for txt files</u>	▪ <u>Extends InfoWriter.</u>
Class Name:TXTLogWriter	
Responsibilities	Collaborations
▪ <u>Writes the lof for txt files.</u>	▪ <u>Extends LogWriter.</u>
Class Name:XMLFileReader	
Responsibilities	Collaborations
<u>Reads file of specific file xml</u>	▪ <u>Extends FileReader.</u>

Class Name:XMLInfoWriter	
Responsibilities	Collaborations
<u>Writes the info file for xml files.</u>	▪ <u>Extends InfoWriter.</u>

Class Name:XMLLogWriter	
Responsibilities	Collaborations
<u>Writes the log file for xml file types.</u>	▪ <u>Extends LogWriter</u>
Class Name:Address	
Responsibilities	Collaborations
<u>Has informtaion for address object.</u>	

Class Name:Company	
Responsibilities	Collaborations
<u>Informtions for companys object.</u>	

Class Name:Date	
Responsibilities	Collaborations
<u>Informtions for date object.</u>	

Class Name:TaxPayer	
Responsibilities	Collaborations
<u>Informtions for tax payer objects.</u> <u>Abstract class.</u>	<u>Extended by SingleTaxpayer,</u> <u>MarriedFilingSeparatelyTaxpayer,</u> <u>MarriedFilingJointlyTaxpayer and</u> <u>HeadOfHouseholdTaxpayer.</u>

Class Name:Receipt	
Responsibilities	Collaborations
<u>Informtions for receipt objects.</u>	

Class Name:TaxPayerFactory	
Responsibilities	Collaborations
<u>Factory class for tax payers.</u>	

Class Name:ReceiptAlreadyExsistsException
--

Responsibilities	Collaborations
<u>Exception if a receipt the user wants to add already exists</u>	

Class Name:WrongFileEndingException	
Responsibilities	Collaborations
<u>If the file type is not supported exception.</u>	

Class Name:WrongFileFormatException	
Responsibilities	Collaborations
<u>If the file type is not supported exception.</u>	

Class Name:WrongReceiptDateException	
Responsibilities	Collaborations
<u>Exception if the receipt date is incorect.</u>	

Class Name:WrongReceiptKindException	
Responsibilities	Collaborations
<u>Exception for kind witch is not supported.</u>	

Class Name:WrongTaxPayerStatusException	
Responsibilities	Collaborations
<u>Exception for wrong status of the tax payer.</u>	

Class Name:ChartDisplay	
Responsibilities	Collaborations
<u>Class for display charts for the tax payer data.</u>	

Class Name:GraphicalInterface	
Responsibilities	Collaborations
<u>GUI for the application uses singleton design pattern.</u>	

Class Name:NewReceipt	
Responsibilities	Collaborations
<u>This class open a new custon JDialog for adding new receipts.</u>	

