
<TAX CALCULATOR>

OVERALL REPORT

IOANNINA 15/12/2022

ALEXANDROS NIKOLAOU 4126

KONSTANTINOS MOUROUSIDIS 4114

TABLE OF CONTENTS

Introduction	4
Refactored Design	4
Use Cases	4
Architecture	10
Detailed Design	11
Classes Responsibilities and Collaborations (CRC CARDS)	15

INTRODUCTION

The main objectives of this phase are to refactor the application so we can improve the efficiency, specify the use cases, make some improvements tasks and test the application with JUnit tests. We display the UML diagrams, too.

REFACTORED DESIGN

USE CASES

LoadTaxPayer

Use case ID	<u>1</u>
Actors	<u>User</u>
Pre conditions	<u>The user has to open the application</u>
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the user click on the "Load Taxpayer" button2. A new window appears and the user has to<ol style="list-style-type: none">2.1. type the tax registration number2.2. choose the type of the file he wants to load(txt or xml)2.3. click OK button
Alternative flow 1	<u>If the user typed a non valid tax registration number or a registration number that not exists , an error message will appear</u>
Alternative flow 2	<u>If the user click on Cancel button, the application returns to the main frame</u>
Post conditions	<u>The application shows the tax payer's number on the screen and loads his data</u>

SelectTaxPayer

Use case ID	<u>2</u>
Actors	<u>User</u>
Pre conditions	<u>The user must have loaded some tax-payers' data</u>
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the user click on the "Select Taxpayer" button 2. A new window appears and the user has to <ol style="list-style-type: none"> 2.1. type the tax registration number 2.2. click OK button
Alternative flow 1	<u>If the user typed a non valid tax registration number or a registration number that not exists , an error message will appear</u>
Alternative flow 2	<u>If the user click on Cancel button, the application returns to the main frame</u>
Post conditions	<u>The application shows his data and return to the main frame</u>

DeleteTaxPayer

Use case ID	<u>3</u>
Actors	<u>User</u>
Pre conditions	<u>The user must have loaded some tax-payers' data</u>
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the user click on the "Delete Taxpayer" button 2. A new window appears with a list with the loaded tax payers and the user has to <ol style="list-style-type: none"> 2.1. type the tax registration number or to click on to the tax payer he wants to delete

	2.2. click delete button
Alternative flow 1	<u>If the user typed a non valid tax registration number or a registration number that not exists , an error message will appear</u>
Alternative flow2	<u>If the user click on Cancel button, the application returns to the main frame</u>
Post conditions	<u>The application deletes the tax payer's data of the programs memory and return to the main frame</u>

SelectTaxPayerByClickOn

Use case ID	<u>4</u>
Actors	<u>User</u>
Pre conditions	<u>The user must have loaded some tax-payers' data</u>
Main flow of events	1. The use case starts when the user click on a tax payer from the list
Post conditions	<u>The application shows his data and return to the main frame</u>

AddReceipt

Use case ID	<u>5</u>
Actors	<u>User</u>

Pre conditions	<u>The user must have select a taxpayer</u>
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the user click on the “Add Receipt” button 2. A new window appears and the user has to complete the fields with the attributes of the receipt he wants to add
Alternative flow 1	<u>If the user typed non valid data an error message will appear</u>
Alternative flow2	<u>If the user click on Cancel button, the application returns to the main frame</u>
Post conditions	<u>The application adds to memory a new receipt</u>

DeleteReceipt

Use case ID	<u>5</u>
Actors	<u>User</u>
Pre conditions	<u>It has to exist receipts of this taxpayer in the memory of the program</u>
Main flow of events	<ol style="list-style-type: none"> 1. The use case starts when the user click on the “Delete Receipt” button 2. A new window appears and the user has to complete the id of the receipt he wants to delete
Alternative flow 1	<u>If the user typed non valid data an error message will appear</u>
Alternative flow2	<u>If the user click on Cancel button, the application returns to the previews frame</u>
Post conditions	<u>The application deletes the receipt from the memory</u>

ViewReport

Use case ID	<u>6</u>
Actors	<u>User</u>
Pre conditions	<u>It has to exist receipts of this taxpayer in the memory of the program</u>
Main flow of events	1. The use case starts when the user click on the “View Report” button
Post conditions	<u>Two new windows appear with graphs from tax payer data</u>

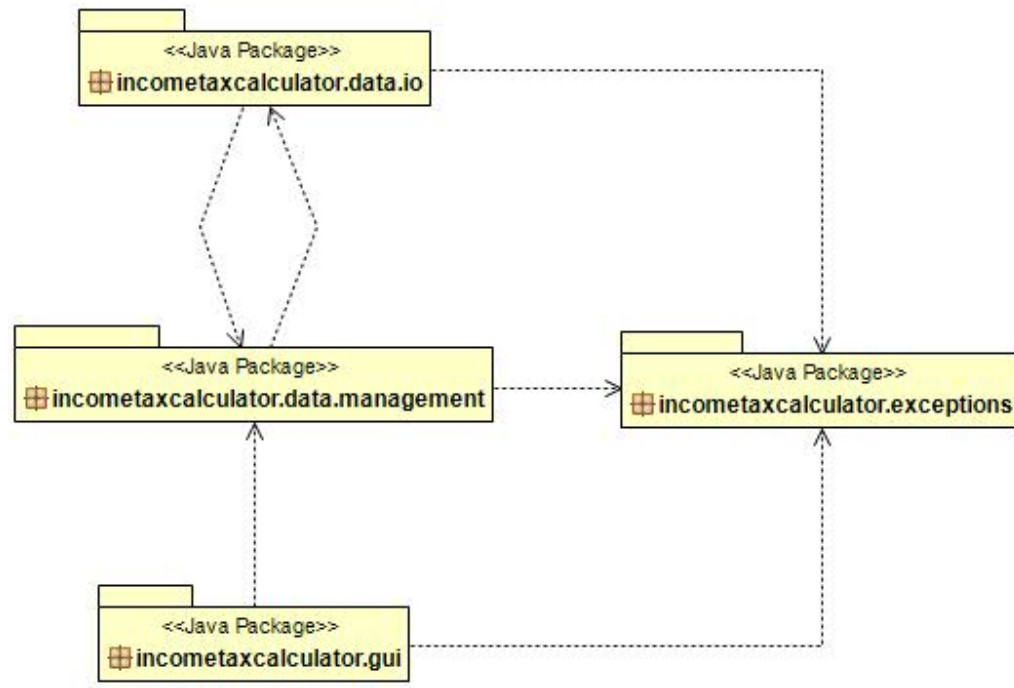
SaveData

Use case ID	<u>7</u>
Actors	<u>User</u>
Pre conditions	<u>It has to exist receipts of this taxpayer in the memory of the program</u>
Main flow of events	1. The use case starts when the user click on the “Save Data” butto 2. The user chooses on what type of file he wants to save the data (xml or txt) 3. Clicks ok button
Alternative flow	1. If the user click on Cancel button, the application returns to the previews frame
Post conditions	<u>A new xml or txt saved in the memory</u>

ShowReceiptByClickOn

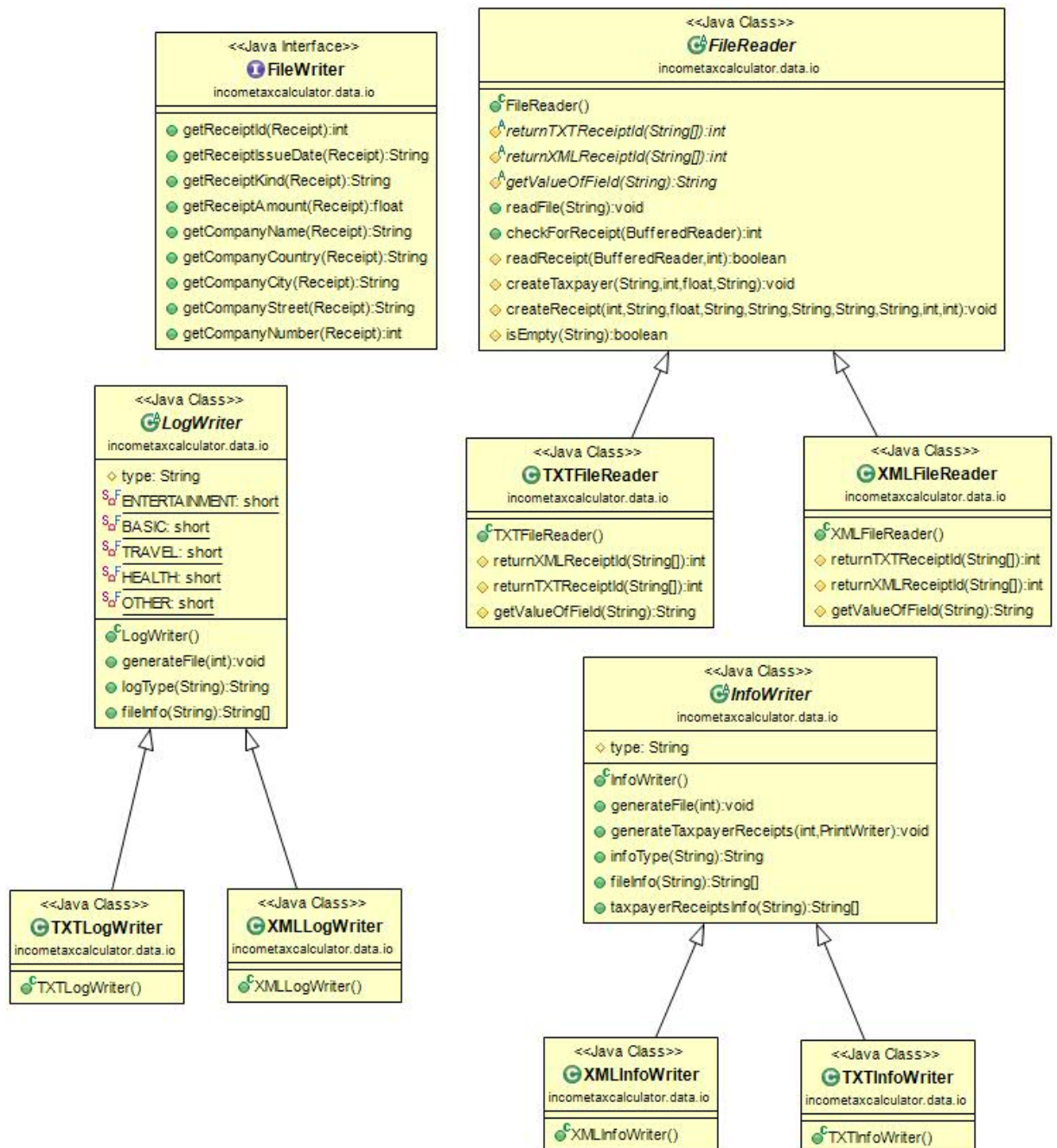
Use case ID	<u>8</u>
Actors	<u>User</u>
Pre conditions	<u>The user must have loaded some tax-payers' data</u>
Main flow of events	<ol style="list-style-type: none">1. The use case starts when the user click on a receipt from the list2. A new window appears with the data of the receipt the user chose3. Click ok to return
Alternative flow	If the user clicks Delete Receipt button the receipt will be delete from system
Post conditions	<u>The application returns to the previews frame</u>

ARCHITECTURE

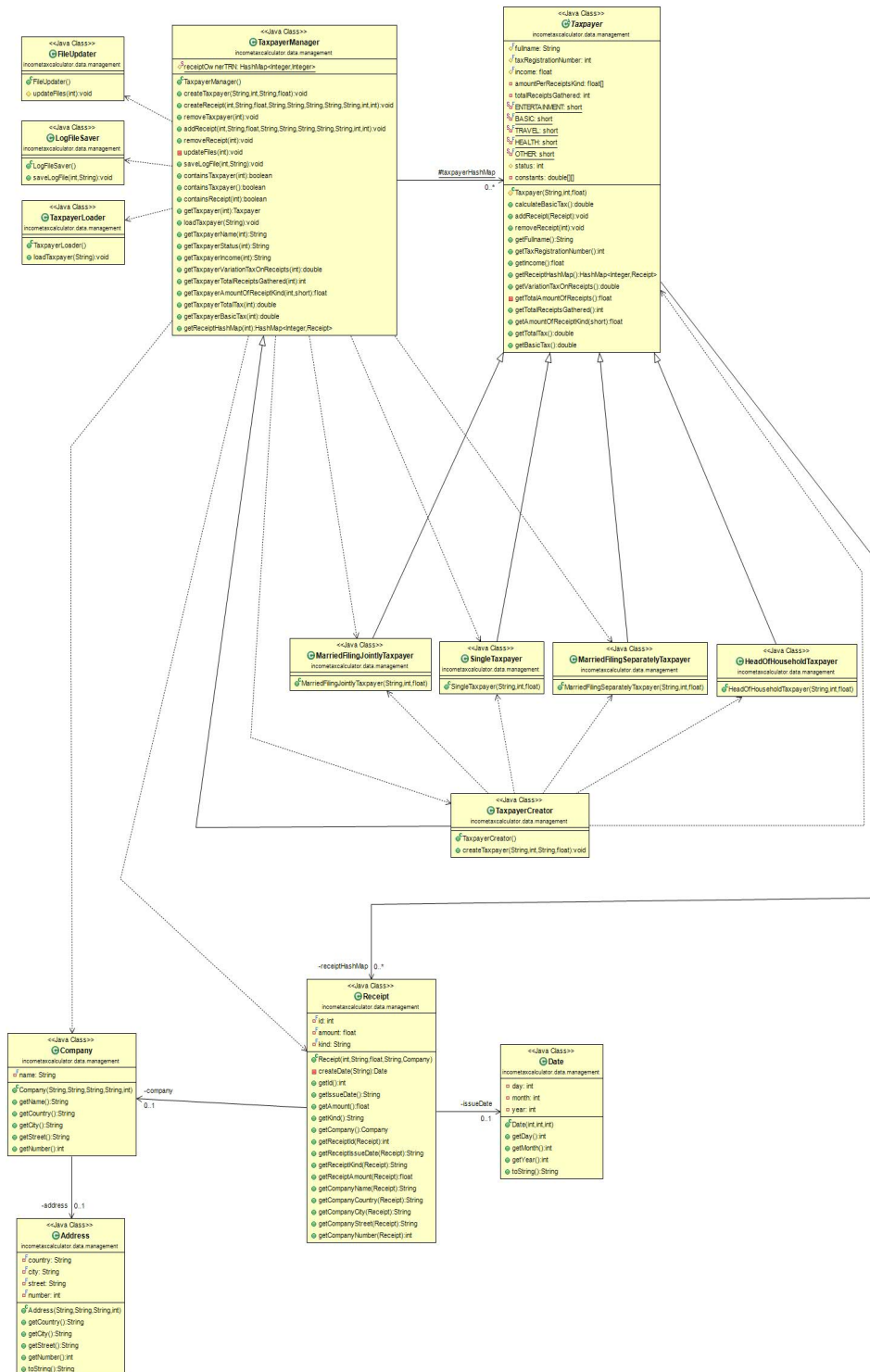


DETAILED DESIGN





- **Package:** incometaxcalculator.data.io











- Package: incometaxcalculator.data.management











- **Package:** incometaxcalculator.data.exceptions





<<Java Class>>	
	WrongReceiptDateException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	WrongReceiptDateException()

<<Java Class>>	
	WrongFileFormatException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	WrongFileFormatException()

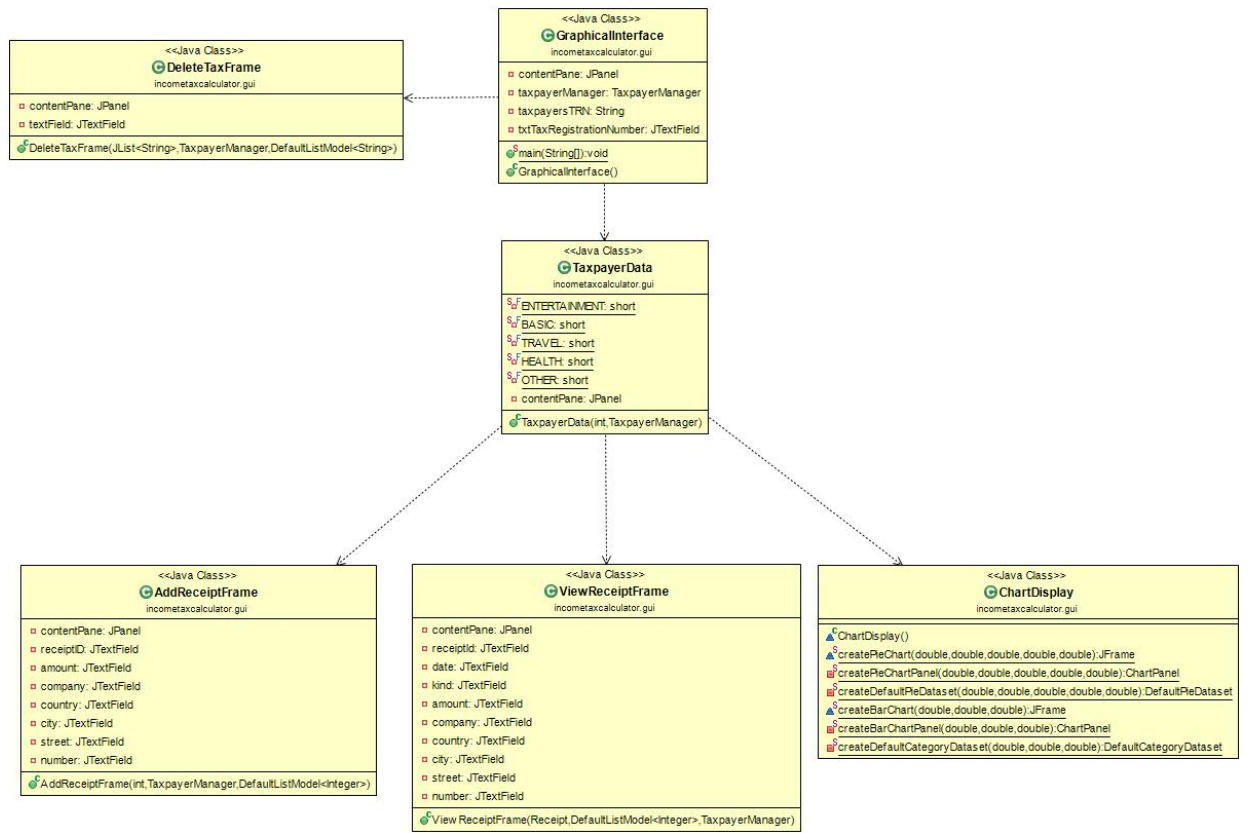
<<Java Class>>	
	WrongFileEndingException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	WrongFileEndingException()

<<Java Class>>	
	WrongReceiptKindException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	WrongReceiptKindException()

<<Java Class>>	
	WrongTaxpayerStatusException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	WrongTaxpayerStatusException()

<<Java Class>>	
	ReceiptAlreadyExistsException
incometaxcalculator.exceptions	
	 <u>serialVersionUID: long</u>
	ReceiptAlreadyExistsException()

- **Package:** incometaxcalculator.data.gui



CLASSES RESPONSIBILITIES AND COLLABORATIONS (CRC CARDS)

Class Name: FileReader	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class performs the reading of the file</u> ▪ <u>This class checks for the existence of receipts of the current taxpayer</u> ▪ <u>This class performs the reading of a receipt</u> ▪ <u>This class can create a Taxpayer</u> ▪ <u>This class can create a receipt</u> 	<ul style="list-style-type: none"> ▪ <u>TaxpayerManager</u>

Class Name: FileWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class is Interface for Receipt class</u> 	<ul style="list-style-type: none"> ▪ <u>Receipt</u>

Class Name: InfoWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class generates a new file</u> ▪ <u>This class generates a receipt for a Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>TaxpayerManager</u> ▪ <u>Receipt</u>

Class Name: LogWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class generates a file</u> 	<ul style="list-style-type: none"> ▪ <u>TaxpayerManager</u>

Class Name:TXTFileReader	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class returns the ReceiptID of a txt file</u> ▪ <u>This class gets values of fields</u> 	<ul style="list-style-type: none"> ▪ <u>File Reader</u>

Class Name: TXTInfoWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class defines the type of a file as txt</u> 	<ul style="list-style-type: none"> ▪ <u>InfoWriter</u>

Class Name: TXTLogWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class defines the type of a file as txt</u> 	<ul style="list-style-type: none"> ▪ <u>LogWriter</u>

Class Name:XMLFileReader	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class returns the ReceiptID of a xml file</u> ▪ <u>This class gets values of fields</u> 	<ul style="list-style-type: none"> ▪ <u>File Reader</u>

Class Name: XMLInfoWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class defines the type of a file as xml</u> 	<ul style="list-style-type: none"> ▪ <u>InfoWriter</u>

Class Name: XMLLogWriter	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class defines the type of a file as xml</u> 	<ul style="list-style-type: none"> ▪ <u>LogWriter</u>

Class Name: Address	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>Handles the information about Address entities</u> 	

Class Name: Company	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>Handles the information about Company entities</u> 	<ul style="list-style-type: none"> ▪ <u>Address</u>

Class Name: Date	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>Handles the information about Date entities</u> 	

Class Name: Receipt	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class handles the information about receipt entities</u> ▪ <u>This class creates new Date objects</u> 	<ul style="list-style-type: none"> ▪ <u>Date</u> ▪ <u>Company</u>

Class Name: Taxpayer	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class handles the information about Taxpayer entities</u> ▪ <u>This class calculates the basic tax</u> ▪ <u>This class handles the receipts of taxpayers</u> 	<ul style="list-style-type: none"> ▪ <u>Receipt</u>

Class Name: FileUpdater	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class updates the files</u> 	<ul style="list-style-type: none"> ▪ <u>XMLInfoWriter</u> ▪ <u>TXTXInfoWriter</u>

Class Name: LogFileSaver	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class saves the files</u> 	<ul style="list-style-type: none"> ▪ <u>XMLInfoWriter</u> ▪ <u>TXTXInfoWriter</u>

Class Name: HeadofHouseholdTaxpayer	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class extends Taxpayer, defines the status of a Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>Taxpayer</u>

Class Name: MarriedFilingJointlyTaxpayer	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class extends Taxpayer, defines the status of a Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>Taxpayer</u>

Class Name: MarriedFilingSeparatelyTaxpayer	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class extends Taxpayer, defines the status of a Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>Taxpayer</u>

Class Name: SingleTaxpayer	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class extends Taxpayer, defines the status of a Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>Taxpayer</u>

Class Name:TaxpayerCreator	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class extends TaxpayerManager</u> ▪ <u>This class creates the requested type of Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>TaxpayerManager</u> ▪ <u>Taxpayer</u> ▪ MarriedFilingJointlyTaxpayer ▪ MarriedFilingSeparatelyTaxpayer ▪ SingleTaxpayer ▪ HeadofHouseholdTaxpayer

Class Name: TaxpayerLoader	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class loads Taxpayer's data from a file</u> 	<ul style="list-style-type: none"> ▪ <u>TaxpayerManager</u> ▪ <u>TXTFileReader</u> ▪ <u>XMLFileReader</u>

Class Name: TaxpayerManager	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class creates and handles object of the classes Receipt, Taxpayer, FileUpdater and the information of Taxpayer</u> 	<ul style="list-style-type: none"> ▪ <u>Receipt</u> ▪ <u>Taxpayer</u> ▪ <u>FileUpdater</u> ▪ <u>TaxpayerLoader</u> ▪ <u>LogFileSaver</u> ▪ <u>TaxpayerCreator</u> ▪ <u>Company</u> ▪ <u>MarriedFilingJointlyTaxpayer</u>

	<ul style="list-style-type: none"> ▪ <u>SingleTaxpayer</u> ▪ <u>MarriedFilingSeparatelyTaxpayer</u>
--	---

Class Name: ReceiptAlreadyExistsException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if a receipt already exists.</u> 	

Class Name: WrongFileEndingException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if the file has wrong ending.</u> 	

Class Name: WrongFileFormatException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if the file has wrong format.</u> 	

Class Name: WrongReceiptDateException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if the date has wrong format.</u> 	

Class Name: WrongReceiptKindException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if the receipt has wrong kind.</u> 	

Class Name: WrongTaxpayerStatusException	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the exception if the Taxpayer has wrong status.</u> 	

Class Name: GraphicalInterface	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the main frame of the app.</u> 	<ul style="list-style-type: none"> ▪ <u>DeleteTaxFrame</u> ▪ <u>TaxpayerData</u>

Class Name: TaxpayerData	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the frame with the data of the taxpayer and other processes.</u> 	<ul style="list-style-type: none"> ▪ <u>AddReceiptFrame</u> ▪ <u>ViewReceiptFrame</u> ▪ <u>ChartDisplay</u>

Class Name: DeleteTaxFrame	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the frame where we can delete loaded taxpayers.</u> 	

Class Name: AddReceiptFrame	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the frame where we can add a receipt into taxpayer data.</u> 	

Class Name: ChartDisplay	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the frame with two different charts of data.</u> 	

Class Name: ViewReceiptFrame	
Responsibilities	Collaborations
<ul style="list-style-type: none"> ▪ <u>This class shows the frame where we can see details of a receipt and delete it if we want.</u> 	