Tools & Techniques

Operating System	:	Windows 10 64bit
Tools	:	Java Development Kit (JDK) 1.8.0_77
Environment	:	Java Runtime Environment (JRE) 1.8.0_77
Back End	:	Java
Front End	:	Terminal/Console App
IDE	:	Netbeans 8.2
Libraries	:	JavaDoc for Documentation, jUnit for Tesing

Console Data

Portfoliot 1

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
JOHN
Enter the reference number
JO001B
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
2
Select the package
1. Small
2. Medium
3. Large
1
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
1
Do you want to choose international calls
Yes
No
No

Portfoliot 2

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
MARY
Enter the reference number
MY001C
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
1
Select the package
1. Small
2. Medium
3. Large
1
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
1
Do you want to choose international calls
Yes
No
NO

Portfoliot 3

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
KEVIN
Enter the reference number
KV001N
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
3
Select the package
1. Small
2. Medium
3. Large
2
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
3
Do you want to choose international calls
Yes
No
NO

Portfoliot 4

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
JOSHUA
Enter the reference number
JS001B
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
4
Select the package
1. Small
2. Medium
3. Large
2
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
4
Do you want to choose international calls
Yes
No
YES

Portfoliot 5

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
PAUL
Enter the reference number
PL002B
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
4
Select the package
1. Small
2. Medium
3. Large
3
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
2
Do you want to choose international calls
Yes
No
YES

Portfoliot 6

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
1
Enter the client name
SARAH
Enter the reference number
SR005N
Select the period
1. One Month
2. 12 Months
3. 18 Months
4. 24 Months
2
Select the package
1. Small
2. Medium
3. Large
3
Select the data bundle
1. Low
2. Medium
3. High
4. Unlimited
4
Do you want to choose international calls
Yes
No
NO
1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit

Summary of Contracts

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
2
Total Number of contracts: 41
Contracts with High or Unlimited Data Bundles: 24
Average charge for large packages: 1493.375
Number of contracts per Month:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	0	0	0	6	5	5	5	5	5	5	5

Summary of Contracts for Specific Month

1. Enter new Contract
2. Display Summary of Contracts
3. Display Summary of Contracts for Selected Month
4. Find and Display Contract
5. Modify existing Contract
0. Exit
3
Select the month from the following
1. Jan
2. Feb
3. Mar
4. Apr
5. May
6. Jun
7. Jul
8. Aug
9. Sep
10. Oct
11. Nov
12. Dec
5
Total Number of contracts: 6
Contracts with High or Unlimited Data Bundles: 3
Average charge for large packages: 1493.375

Contract.txt Data

JOHN JO001B 12 1 1 No 450.0 5400.0 10 13-May-2021
MARY MY001C 1 1 1 NO 350.0 350.0 30 13-May-2021
KEVIN KV001N 18 2 3 NO 997.5 17955.0 5 13-May-2021
JOSHUA JS001B 24 2 4 YES 0.0 0.0 10 13-May-2021
PAUL PL002B 24 3 2 YES 1086.75 26082.0 10 13-May-2021
SARAH SR005N 12 3 4 NO 1900.0 22800.0 5 13-May-2021

Test Files

CreateContractTest

```
package contractmanager.maincode;

import org.junit.Before;
import org.junit.Test;
import static org.junit.Assert.*;

public class CreateContractTest {

   private CreateContract contract;

   @Before
   public void setUp() {
      contract = new CreateContract();
   }

   @Test
   public void testGetCurrentDate() {
      assertNotNull(CreateContract.getDate());
   }

}
```

CreateContractTest Console Output

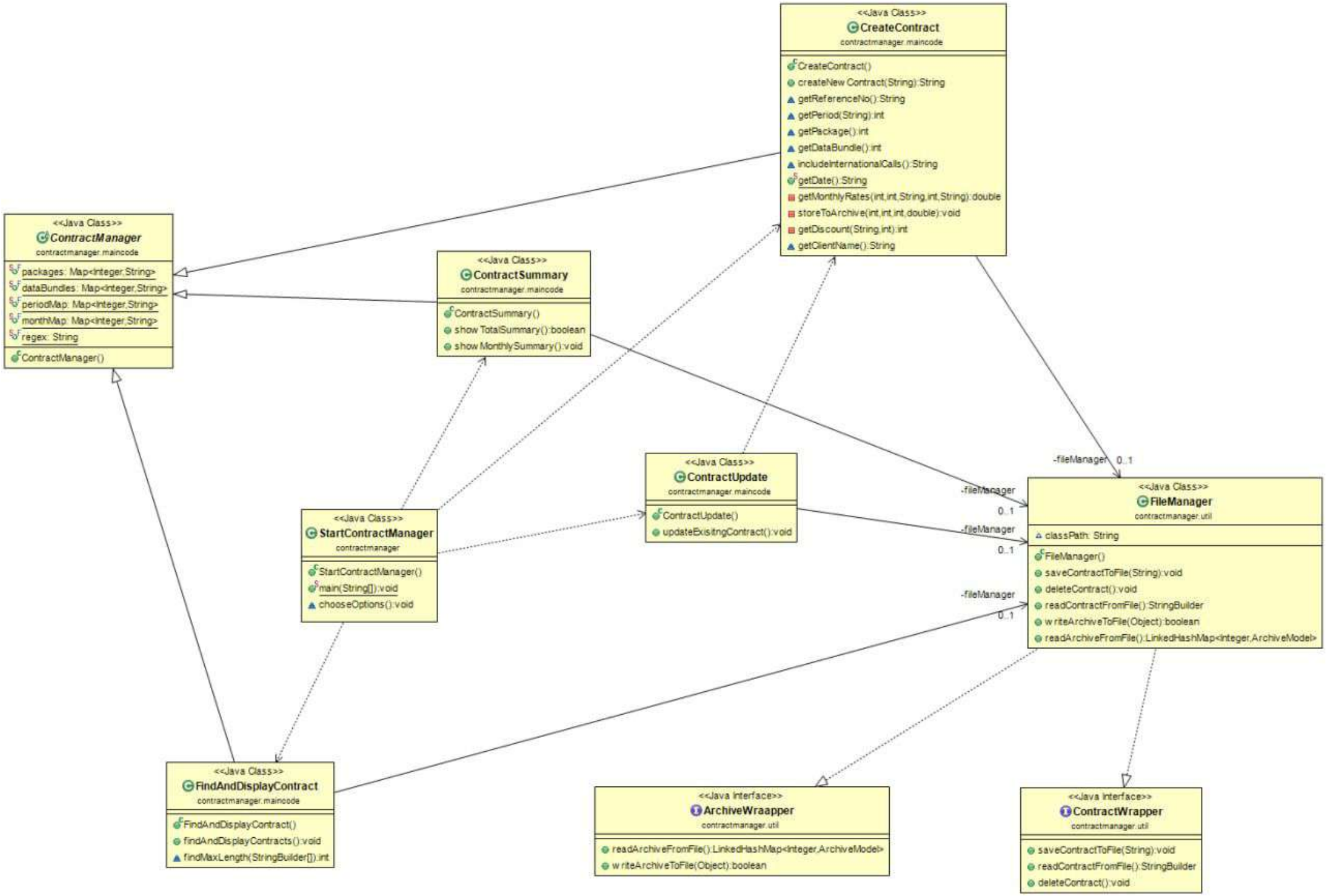
```
Testsuite: contractmanager.maincode.CreateContractTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.038 sec
```

FindAndDisplayContractTest

```
package contractmanager.maincode;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

public class FindAndDisplayContractTest {

   private FindAndDisplayContract displayContract;

   @Before
   public void setUp() {
      displayContract = new FindAndDisplayContract();
   }

   @Test
   public void testFindMaxLength() {
      assertEquals(25, displayContract.findMaxLength(new StringBuilder("Data Of length 17"),
            new StringBuilder("Another Data Of length 25")));
   }

   @Test
   public void testFindMaxLengthZero() {
      assertEquals(0, displayContract.findMaxLength(new StringBuilder("")));
   }

   @Test
   public void testFindMaxLengthOne() {
      assertEquals(1, displayContract.findMaxLength(new StringBuilder(""), new StringBuilder("A")));
   }
}
```

FindAndDisplayContractTest Console Output

```
Testsuite: contractmanager.maincode.FindAndDisplayContractTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.039 sec
```


<<Java Class>>
CreateContract
contractmanager.maincode
CreateContract()
createNewContract(String):String
getReferenceNo():String
getPeriod(String):int
getPackage():int
getDataBundle():int
includeInternationalCalls():String
getDate():String
getMonthlyRates(int,int,String,int,String):double
storeToArchive(int,int,int,double):void
getDiscount(String,int):int
getClientName():String
<<Java Class>>
ContractManager
contractmanager.maincode
packages: Map<Integer,String>
dataBundles: Map<Integer,String>
periodMap: Map<Integer,String>
monthMap: Map<Integer,String>
regex: String
ContractManager()
<<Java Class>>
ContractSummary
contractmanager.maincode
ContractSummary()
showTotalSummary():boolean
showMonthlySummary():void
<<Java Class>>
ContractUpdate
contractmanager.maincode
ContractUpdate()
updateExisitngContract():void
<<Java Class>>
StartContractManager
contractmanager
StartContractManager()
main(String[]):void
chooseOptions():void
<<Java Class>>
FileManager
contractmanager.util
classPath: String
FileManager()
saveContractToFile(String):void
deleteContract():void
readContractFromFile():StringBuilder
writeArchiveToFile(Object):boolean
readArchiveFromFile():LinkedHashMap<Integer,ArchiveModel>
<<Java Class>>
FindAndDisplayContract
contractmanager.maincode
FindAndDisplayContract()
findAndDisplayContracts():void
findMaxLength(StringBuilder[]):int
<<Java Interface>>
ArchiveWraapper
contractmanager.util
readArchiveFromFile():LinkedHashMap<Integer,ArchiveModel>
writeArchiveToFile(Object):boolean
<<Java Interface>>
ContractWrapper
contractmanager.util
saveContractToFile(String):void
readContractFromFile():StringBuilder
deleteContract():void
-fileManager
0..1

ContactManagerDocumentation

Packages

Package	Description
contractmanager	
contractmanager.maincode	
contractmanager.model	
contractmanager.util	

All Classes

ArchiveModel
ArchiveWrapper
ContractManager
ContractSummary
ContractUpdate
ContractWrapper
CreateContract
FileManager
FindAndDisplayContract
InitContractManager

All Classes

Classes

ContractManager

Classes

ContractSummary
ContractUpdate
CreateContract
FindAndDisplayContract
InitContractManager

Classes

ArchiveModel

Interfaces

ArchiveWrapper
ContractWrapper

Classes

FileManager

contractmanager.model

Class ArchiveModel

java.lang.Object
    contractmanager.model.ArchiveModel

All Implemented Interfaces:

`java.io.Serializable`

```
public class ArchiveModel
extends java.lang.Object
implements java.io.Serializable
```

A Class Model/Bean for initializing Contract Manager.

Version:

`3.0`

Author:

`Ash`

See Also:

`Serialized Form`

Constructor Summary

Constructors

Constructor and Description
`ArchiveModel()`

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
long	**getTotalHighOrUnlimited**() default value
long	**getTotalLargePackage**() default value
double	**getTotalLargePackageCharge**() default value
long	**getTotalNo**() default value
void	**setTotalHighOrUnlimited**(long totalHighOrUnlimited)
void	**setTotalLargePackage**(long totalLargePackage)
void	**setTotalLargePackageCharge**(double totalLargePackageCharge)
void	**setTotalNo**(long totalNo)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ArchiveModel

```
public ArchiveModel()
```

Method Detail

getTotalNo

```
public long getTotalNo()
```

default value

Returns:

```
totalNo (in long)
```

setTotalNo

```
public void setTotalNo(long totalNo)
```

Parameters:

```
ArchiveModel - #totalNo to set (in long)
```

getTotalHighOrUnlimited

```
public long getTotalHighOrUnlimited()
```

default value

Returns:

```
totalHighOrUnlimited (in )
```

setTotalHighOrUnlimited

```
public void setTotalHighOrUnlimited(long totalHighOrUnlimited)
```

Parameters:

```
ArchiveModel - #totalHighOrUnlimited to set (in long)
```

getTotalLargePackageCharge

```
public double getTotalLargePackageCharge()
```

default value

Returns:

totalLargePackageCharge (in double)

setTotalLargePackageCharge

```
public void setTotalLargePackageCharge(double totalLargePackageCharge)
```

Parameters:

ArchiveModel - #totalLargePackageCharge to set (in double)

getTotalLargePackage

```
public long getTotalLargePackage()
```

default value

Returns:

totalLargePackage (in long)

setTotalLargePackage

```
public void setTotalLargePackage(long totalLargePackage)
```

Parameters:

ArchiveModel - #totalLargePackage to set (in long)

contractmanager.util

Interface ArchiveWrapper

All Known Implementing Classes:

FileManager

---

public interface **ArchiveWrapper**

The ArchiveWrapper is an Interface for Archive Operations Archive Operations specifically includes file operations regarding archive entries for Contract Manager

Version:

3.0

Author:

Ash

Method Summary

All Methods | Instance Methods | Abstract Methods

Modifier and Type	Method and Description
java.util.LinkedHashMap<java.lang.Integer,ArchiveModel>	**readArchiveFromFile**() ArchiveWrapper implementation in FileManager
boolean	**writeArchiveToFile**(java.lang.Object object) ArchiveWrapper implementation in FileManager

Method Detail

readArchiveFromFile

java.util.LinkedHashMap<java.lang.Integer,ArchiveModel> readArchiveFromFile()

ArchiveWrapper implementation in FileManager

Returns:

null

See Also:

Integer, ArchiveModel, FileManager, #readArchiveFromFile()

writeArchiveToFile

boolean writeArchiveToFile(java.lang.Object object)

ArchiveWrapper implementation in FileManager

Parameters:

object - data to be saved to archive repository(file)

Returns:

null

See Also:

FileManager, #writeArchiveToFile(Object)

contractmanager

Class ContractManager

java.lang.Object
    contractmanager.ContractManager

```
public class ContractManager
extends java.lang.Object
```

The Main Class for Contract Manager implements an Application/Prototype the Telephone Contract Manager.

Version:

3.0

Author:

Ash

Constructor Summary

Constructors

Constructor and Description
`ContractManager()`

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
void	`chooseOptions()` displays the Contract options 1.

static void	**main**(java.lang.String[] args) main method for Contract Manager, Deployment/execution of ContractManager

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ContractManager

```
public ContractManager()
```

Method Detail

main

```
public static void main(java.lang.String[] args)
```

main method for Contract Manager, Deployment/execution of ContractManager

Parameters:

`args` - Unused

Throws:

`java.util.InputMismatchException` - On input error/Any exception

See Also:

`ContractManager()`, `chooseOptions()`

chooseOptions

```
public void chooseOptions()
```

displays the Contract options 1. Enter new Contract

Throws:

java.util.InputMismatchException - On input error/Any exception

See Also:

#createNewContract(String) 2. Display Summary of Contracts, #showTotalSummary() 3. Display Summary of Contracts for Selected Month, #showMonthlySummary() 4. Find and Display Contract, #findAndDisplayContracts() 5. Modify existing Contract, #updateExisitngContract() 0. Exit, #exit(int) The user selected option is assigned to the variable input via Scanner, then executes the corresponding operations if a valid selection if not the control execute the same operations by traversing the current method along the infinitely

contractmanager.maincode

Class ContractSummary

java.lang.Object
  contractmanager.maincode.InitContractManager
    contractmanager.maincode.ContractSummary

```
public class ContractSummary
extends InitContractManager
```

The Class ContractSummary provides the user about Summary of Contracts in our repositories ContractSummary to retrieve the - Total number of contracts - Number of contracts with High Data Bundle - Number of contracts with Unlimited Data Bundle - Average charge of large package contracts - Total number of contracts per month (assuming the files only include data for the current year only) Also show an entry for all twelve months even those that have an entry of zero

Version:

3.0

Author:

Ash

Field Summary

Fields inherited from class contractmanager.maincode.InitContractManager

DATA_BUNDLES, FILE_MANAGER, MONTH_MAP, PACKAGES, PERIOD_MAP, REGEX, scanner

Constructor Summary

Constructors

Constructor and Description

ContractSummary()

Method Summary

All Methods	Instance Methods	Concrete Methods

Modifier and Type	Method and Description
void	**showMonthlySummary**() To display just the Summary of a month user is asked for an integer input that corresponds to a month of year The user selected option is assigned to the variable month via Scanner if not a valid month is selected, the control execute the same operations by traversing the same method @see #showMonthlySummary() but on a valid month selection, it fetches yearly data from the archive repository then fetches the specific month data from the yearly data using the integer month value which maps to the index to the month in the yearly data
void	**showTotalSummary**() Display Summary of Contracts - Total Number of contracts - Number of Contracts with High or Unlimited Data Bundles - Average charge of large package contracts - Total number of contracts per month Fetches yearly data from the archive repository

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ContractSummary

```
public ContractSummary()
```

Method Detail

showTotalSummary

```
public void showTotalSummary()
```

Display Summary of Contracts - Total Number of contracts - Number of Contracts with High or Unlimited Data Bundles - Average charge of large package contracts - Total number of contracts per month Fetches yearly data from the archive repository

See Also:

`#readArchiveFromFile(), InitContractManager.MONTH_MAP, ArchiveModel, #getTotalNo(), #getTotalHighOrUnlimited(), #getTotalLargePackage(), #getTotalLargePackageCharge() segregates the yearly data to the corresponding months from jan the start to the current month of the year then executes the computation of Average charge for large packages`

showMonthlySummary

```
public void showMonthlySummary()
```

To display just the Summary of a month user is asked for an integer input that corresponds to a month of year The user selected option is assigned to the variable month via Scanner if not a valid month is selected, the control execute the same operations by traversing the same method @see #showMonthlySummary() but on a valid month selection, it fetches yearly data from the archive repository then fetches the specific month data from the yearly data using the integer month value which maps to the index to the month in the yearly data

See Also:

`#readArchiveFromFile(), InitContractManager.MONTH_MAP, ArchiveModel, #getTotalNo(), #getTotalHighOrUnlimited(), #getTotalLargePackage(), #getTotalLargePackageCharge() then executes the computation of Average charge for large packages`

contractmanager.maincode

Class ContractUpdate

java.lang.Object
  contractmanager.maincode.InitContractManager
    contractmanager.maincode.ContractUpdate

```
public class ContractUpdate
extends InitContractManager
```

The Class ContractUpdate provides the client contract reference and their whole details user gets to search and fetch the contract details and then if needed, user can update the contract details but only if that contract meets the company's predetermined criteria user can search for the contract details and fetch the details

Version:

3.0

Author:

Ash

Field Summary

Fields inherited from class contractmanager.maincode.InitContractManager

DATA_BUNDLES, FILE_MANAGER, MONTH_MAP, PACKAGES, PERIOD_MAP, REGEX, scanner

Constructor Summary

Constructors

Constructor and Description

ContractUpdate()

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
void	updateExisitngContract() To update any contract user has to find the contract first, henceforth the user is asked for a Search keyword to find the contract that he intends to update.

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ContractUpdate

```
public ContractUpdate()
```

Method Detail

updateExisitngContract

```
public void updateExisitngContract()
```

To update any contract user has to find the contract first, henceforth the user is asked for a Search keyword to find the contract that he intends to update. the input is assigned to the variable searchTerm via Scanner

See Also:

#readContractFromFile(), #deleteContract(), #saveContractToFile(String), #createNewContract(String) this searchTerm originally meant for a contract file and it is then cross checked and searched for any matches (full or partial) against the reference and client name within our contracts repository if found multiple

matches the user is listed with all the matches found and asked to choose the contract he/she intends to update The user selected option is assigned to the variable choice via Scanner if not a valid choice is selected, the control execute the same operations by traversing the same method but on a valid contract choice selection, it fetches the corresponding contract data from the already stored contract repository that matches the searchTerm then fetches the specific month data from the yearly data using the integer month value which maps to the index to the month in the yearly data then executes the computation of Average charge for large packages

contractmanager.util

Interface ContractWrapper

All Known Implementing Classes:

`FileManager`

`public interface` **`ContractWrapper`**

The ContractWrapper is an Interface for Contract Operations, specifically file operations regarding contract entries for Contract Manager

Version:

`3.0`

Author:

`Ash`

Method Summary

All Methods | **Instance Methods** | **Abstract Methods**

Modifier and Type	Method and Description
`void`	**`deleteContract`**`()` ContractWrapper implementation in FileManager
`java.lang.StringBuilder`	**`readContractFromFile`**`()` ContractWrapper implementation in FileManager
`void`	**`saveContractToFile`**`(java.lang.String data)` ContractWrapper implementation in FileManager

Method Detail

saveContractToFile

```
void saveContractToFile(java.lang.String data)
```

ContractWrapper implementation in FileManager

Parameters:

`data - String data to saved to contract repository(file)`

See Also:

`String, FileManager, #saveContractToFile(String)`

readContractFromFile

```
java.lang.StringBuilder readContractFromFile()
```

ContractWrapper implementation in FileManager

See Also:

`FileManager, #readContractFromFile() Contract Data in StringBuilder object(type), StringBuilder`

deleteContract

```
void deleteContract()
```

ContractWrapper implementation in FileManager

See Also:

`FileManager, #deleteContract()`

contractmanager.maincode

Class CreateContract

java.lang.Object
    contractmanager.maincode.InitContractManager
        contractmanager.maincode.CreateContract

```
public class CreateContract
extends InitContractManager
```

Class CreateContract for initializing new Contracts for the Contract Manager

Version:

3.0

Author:

Ash

Field Summary

Fields inherited from class contractmanager.maincode.InitContractManager

DATA_BUNDLES, FILE_MANAGER, MONTH_MAP, PACKAGES, PERIOD_MAP, REGEX, scanner

Constructor Summary

Constructors

Constructor and Description
CreateContract()

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
java.lang.String	**createNewContract**(java.lang.String referenceNo) createNewContract triggers the creation of a new contract
java.lang.String	**getClientName**() user is asked to enter a Client Name for the creation of new contract the input is assigned to the variable clientName via Scanner if clientName exceeds the letter limit validation, the control execute the same operations by traversing the same method
int	**getDataBundle**() user is asked to choose the Data Bundle for the creation of new contract with the options the input is assigned to the variable choice via Scanner choice determines the Data Bundle if invalid choice for Data Bundle is selected, the control execute the same operations by traversing the same method
static java.lang.String	**getDate**()
int	**getDiscount**(java.lang.String referenceNum, int period)
double	**getMonthlyRates**(int dataPackage, int dataBundle, java.lang.String referenceNum, int period, java.lang.String intlCalls)
int	**getPackage**() user is asked to choose the Package for the creation of new contract with the options the input is assigned to the variable choice via Scanner choice determines the package if invalid choice for package is selected, the control execute the same operations by traversing the same method
int	**getPeriod**(java.lang.String referenceNum)
java.lang.String	**getReferenceNo**() user is asked for reference number for the creation of new contract the input is assigned to the variable referenceNum via Scanner referenceNum is validated w.r.t regex referenceNum @see String#matches(String) if validation is failed, the control execute the same operations by traversing the same method and if validation is successful it is checked for redundancy
java.lang.String	**includeInternationalCalls**() user is asked to choose whether to include the International calls for this contract on process with simple Yes/No options the input is assigned to the variable choice via Scanner choice determines whether to include the International calls for this contract on process if invalid choice for Data Bundle is selected, the

	control execute the same operations by traversing the same method
void	**storeToArchive**(int contractPackage, int dataBundle, int period, double monthlyRates)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

CreateContract

public CreateContract()

Method Detail

createNewContract

public java.lang.String createNewContract(java.lang.String referenceNo)

createNewContract triggers the creation of a new contract

Parameters:

referenceNo - reference number is null for creating new contract and have value if updating contract

Returns:

Contract details concatenated to a string object and returned

See Also:

#getClientName(), #getReferenceNo(), #getPeriod(), #getPackage(), #getDataBundle(), #includeInternationalCalls(), getDiscount(String, int), getMonthlyRates(int, int, String, int, String), getDate(), #readContractFromFile(), #saveContractToFile(String), storeToArchive(int, int, int, double)

getReferenceNo

```
public java.lang.String getReferenceNo()
```

user is asked for reference number for the creation of new contract the input is assigned to the variable referenceNum via Scanner referenceNum is validated w.r.t regex referenceNum @see String#matches(String) if validation is failed, the control execute the same operations by traversing the same method and if validation is successful it is checked for redundancy

Returns:

`referenceNo of type string object`

See Also:

`#readContractFromFile() again, if validation is failed, the control execute the same operations by traversing the same method and if no referenceNum exists like now it returns the referenceNum to createNewContract, createNewContract(String)`

getPeriod

```
public int getPeriod(java.lang.String referenceNum)
```

Parameters:

`referenceNum - is required to make sure Business Customers chooses right period for the contract as per company policies user is asked to choose the period for the creation of new contract with the options the input is assigned to the variable choice via Scanner choice determines the period which is validated as per the policies of the company like Business Customers must choose atleast 12 months package if invalid choice is selected, the control execute the same operations by traversing the same method`

Returns:

`period for the new contract as type int`

getPackage

```
public int getPackage()
```

user is asked to choose the Package for the creation of new contract with the options the input is assigned to the variable choice via Scanner choice determines the package if invalid choice for package is selected, the control execute the same operations by traversing the same method

Returns:

`package for the new contract as type int`

getDataBundle

```
public int getDataBundle()
```

user is asked to choose the Data Bundle for the creation of new contract with the options the input is assigned to the variable choice via Scanner choice determines the Data Bundle if invalid choice for Data Bundle is selected, the control execute the same operations by traversing the same method

Returns:

```
Data Bundle choice for the new contract as type int
```

includeInternationalCalls

```
public java.lang.String includeInternationalCalls()
```

user is asked to choose whether to include the International calls for this contract on process with simple Yes/No options the input is assigned to the variable choice via Scanner choice determines whether to include the International calls for this contract on process if invalid choice for Data Bundle is selected, the control execute the same operations by traversing the same method

Returns:

```
choice for includeInternationalCalls for the new contract as type String
```

getDate

```
public static java.lang.String getDate()
```

Returns:

```
date instance formatted to String Type
```

See Also:

```
Calendar.getInstance(), Calendar.getTime(), SimpleDateFormat.SimpleDateFormat(String),
DateFormat.format(java.util.Date)
```

getMonthlyRates

```
public double getMonthlyRates(int dataPackage,
                              int dataBundle,
                              java.lang.String referenceNum,
                              int period,
                              java.lang.String intlCalls)
```

Parameters:

dataPackage - user chosen data package

dataBundle - user chosen data bundle

referenceNum - user given unique reference number

period - user chosen period of contract

intlCalls - user choice, either Yes/No

Returns:

computed rate of type double

See Also:

compute the rate w.r.t the discount according to dataBundle and dataPackage

storeToArchive

```
public void storeToArchive(int contractPackage,
                           int dataBundle,
                           int period,
                           double monthlyRates)
```

Parameters:

contractPackage - user chosen data package

dataBundle - user chosen data bundle

period - user chosen period of contract

monthlyRates - computed rate for the contract

See Also:

Calendar.getInstance(), #readArchiveFromFile(), ArchiveModel, #setTotalNo(long)(), #setTotalHighOrUnlimited(long)(), #setTotalLargePackageCharge(double)(), #setTotalLargePackage(long)(), #writeArchiveToFile(Object)()

getDiscount

```
public int getDiscount(java.lang.String referenceNum,
                       int period)
```

Parameters:

`referenceNum - user given unique reference number`

`period - user chosen period of contract determining customerType from referenceNum @see String#charAt(int) computation of discount`

Returns:

`computed discount amount of Type int`

getClientName

`public java.lang.String getClientName()`

user is asked to enter a Client Name for the creation of new contract the input is assigned to the variable clientName via Scanner if clientName exceeds the letter limit validation, the control execute the same operations by traversing the same method

Returns:

`clientName of type String is formatted and returned`

contractmanager.util

Class FileManager

java.lang.Object
    contractmanager.util.FileManager

All Implemented Interfaces:

`ArchiveWrapper, ContractWrapper`

```
public class FileManager
extends java.lang.Object
implements ContractWrapper, ArchiveWrapper
```

Class FileManager is for Contract Manager's File operations

Version:

`3.0`

Author:

`Ash`

Constructor Summary

Constructors

Constructor and Description
`FileManager()`

Method Summary

All Methods **Instance Methods** **Concrete Methods**

Modifier and Type	Method and Description
void	**deleteContract**() delete Contract details from Contract repository(File) ContractWrapper implementation in FileManager
java.util.LinkedHashMap<java.lang.Integer,**ArchiveModel**>	**readArchiveFromFile**() read Contract details to Archive repository(File) ArchiveWrapper implementation in FileManager
java.lang.StringBuilder	**readContractFromFile**() read Contract details from Contract repository(File) ContractWrapper implementation in FileManager
void	**saveContractToFile**(java.lang.String data) Save/write Contract details into Contract repository(File)
boolean	**writeArchiveToFile**(java.lang.Object object) write Contract details to Archive repository(File) ArchiveWrapper implementation in FileManager

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

FileManager

```
public FileManager()
```

Method Detail

saveContractToFile

```
public void saveContractToFile(java.lang.String data)
```

Save/write Contract details into Contract repository(File)

Specified by:

`saveContractToFile` in interface `ContractWrapper`

Parameters:

`data - String data to saved to contract repository(file)`

See Also:

`ContractWrapper.saveContractToFile(String)`, `classPath`, `#File(String)`, `#exists()`, `#mkdirs()`, `#createNewFile()`, `#FileWriter(File)`, `#write(char[])`, `#flush()`, `#close()`

deleteContract

```
public void deleteContract()
```

delete Contract details from Contract repository(File) ContractWrapper implementation in FileManager

Specified by:

`deleteContract` in interface `ContractWrapper`

See Also:

`ContractWrapper.deleteContract()`, `classPath`, `#File(String)`, `#exists()`, `#delete()`

readContractFromFile

```
public java.lang.StringBuilder readContractFromFile()
```

read Contract details from Contract repository(File) ContractWrapper implementation in FileManager

Specified by:

`readContractFromFile` in interface `ContractWrapper`

Returns:

`null`

See Also:

`ContractWrapper.deleteContract()`, `classPath`, `#File(String)`, `#exists()`, `#FileReader(File)`, `#close()`

writeArchiveToFile

```
public boolean writeArchiveToFile(java.lang.Object object)
```

write Contract details to Archive repository(File) ArchiveWrapper implementation in FileManager

Specified by:

`writeArchiveToFile` in interface `ArchiveWrapper`

Parameters:

`object - data to be saved to archive repository(file)`

Returns:

`false of type boolean`

See Also:

`ArchiveWrapper.writeArchiveToFile(Object)`, `classPath`, `#File(String)`, `#exists()`, `#mkdirs()`, `#createNewFile()`, `#FileOutputStream(File)`, `#ObjectOutputStream(File)`, `#writeObject(Object)`, `#close()`

readArchiveFromFile

```
public java.util.LinkedHashMap<java.lang.Integer,ArchiveModel> readArchiveFromFile()
```

read Contract details to Archive repository(File) ArchiveWrapper implementation in FileManager

Specified by:

`readArchiveFromFile` in interface `ArchiveWrapper`

Returns:

`archive repository stored in file into type LinkedHashMap with index/key of #Integer type and value of type #ArchiveModel object>`

See Also:

`ArchiveWrapper.readArchiveFromFile()`, `classPath`, `#File(String)`, `#exists()`, `#FileInputStream(File)`, `#ObjectInputStream(File)`, `#readObject()`, `#close()`

contractmanager.maincode

Class FindAndDisplayContract

java.lang.Object
    contractmanager.maincode.InitContractManager
        contractmanager.maincode.FindAndDisplayContract

```
public class FindAndDisplayContract
extends InitContractManager
```

Class FindAndDisplayContract enables users to list out the contract data for the Contract Manager

Version:

3.0

Author:

Ash

Field Summary

Fields inherited from class contractmanager.maincode.InitContractManager

DATA_BUNDLES, FILE_MANAGER, MONTH_MAP, PACKAGES, PERIOD_MAP, REGEX, scanner

Constructor Summary

Constructors

Constructor and Description
FindAndDisplayContract()

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
void	**findAndDisplayContracts**() find and display Contracts in the Contract Manager repositories To find and display Contracts any contract, user is asked for a Search keyword/term to find the contract that he intends to find and display
int	**findMaxLength**(java.lang.StringBuilder... str) a variable max is declared as int and assigned to it's data type default value i.e 0 (default value for primitive data type int)

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

FindAndDisplayContract

```
public FindAndDisplayContract()
```

Method Detail

findAndDisplayContracts

```
public void findAndDisplayContracts()
```

find and display Contracts in the Contract Manager repositories To find and display Contracts any contract, user is asked for a Search keyword/term to find the contract that he intends to find and display

See Also:

#readContractFromFile() the Search keyword/term is intended either for unique reference number or client

name of a contract file it is then searched for the match against the reference number and client name within our Contract Manager repositories if found the match right after fetching the contract details, the customerType of the contract has to be determined from the reference number details of the contract are aligned row by row i.e "Client Name", "Reference No and Date", "Package and Data Bundle", "Period and Customer Type", "Discount and International Call Status" and "computed monthly rate" with the help of non primitive data type StringBuilder #StringBuilder the so called alignment achieved by appending spaces to each row, according to the computed highest length of each row's max length - FindAndDisplayContract findMaxLength(StringBuilder...), the summary output to be displayed is of the following format +---+ | | | Customer: J Mason | | | | Ref: JB123B Date: 15 Oct 2017 | | Package: Large (1200) Data: Low (1GB) | | Period: 12 Months Type: Business | | | | Discount: 10% Intl. Calls: No | | | | Discounted Monthly Charge: £7.65 | | | +---+

findMaxLength

`public int findMaxLength(java.lang.StringBuilder... str)`

a variable max is declared as int and assigned to it's data type default value i.e 0 (default value for primitive data type int)

Parameters:

str - - varying array of type StringBuilder str containing the elements of type StringBuilder represents/are the whole rows in the summary display format therefore the highest/max row size is taken for the border and format estimation

Returns:

variable max of type int, which possess the highest row size among the summary rows of type StringBuilder from the varying array str

contractmanager.maincode

Class InitContractManager

java.lang.Object
    contractmanager.maincode.InitContractManager

Direct Known Subclasses:

ContractSummary, ContractUpdate, CreateContract, FindAndDisplayContract

---

```
public abstract class InitContractManager
extends java.lang.Object
```

An Abstract Class for Contract Manager with essential Constants and Fields Initializing/Assigning Constant Fields : REGEX, PACKAGES, DATA_BUNDLES, PERIOD_MAP, MONTH_MAP to : -Create a Contract -Find And Display Contract(s) -Get Summary of a Contract(s) -Modify Existing Contract(s)

Version:

3.0

Author:

Ash

Field Summary

Fields

Modifier and Type	Field and Description
static java.util.Map<java.lang.Integer,java.lang.String>	**DATA_BUNDLES** static Constant assigned for the Data Bundles of the company Data Bundles of the company : -Low (1GB) -Medium (4GB) -High (8GB) -Unlimited
static **FileManager**	**FILE_MANAGER**

	Constant assigned with instance of #FileManager for Contract Manager's file operations
static java.util.Map<java.lang.Integer,java.lang.String>	**MONTH_MAP** static Constant assigned to calculate the total number of contracts per month, assuming the files only include data for the current year only
static java.util.Map<java.lang.Integer,java.lang.String>	**PACKAGES** static Constant assigned for the company offering packages Company providing packages : -Small 300 mins -Medium 600 mins -Large 1200 mins
static java.util.Map<java.lang.Integer,java.lang.String>	**PERIOD_MAP** static Constant assigned for Contract periods as per the policies of the company Contract periods as per the policies of the company : 1, 12, 18 or 24 months
static java.lang.String	**REGEX** static Constant assigned for matching the company's new Reference Number Format Reference Number Format explained : -Two letter followed by three digits and a letter.
static java.util.Scanner	**scanner** static field assigned with instance of #Scanner to get user input for Contract Manager's queries

Constructor Summary

Constructors

Constructor and Description
InitContractManager()

Method Summary

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

scanner

public static java.util.Scanner scanner

static field assigned with instance of #Scanner to get user input for Contract Manager's queries

See Also:

#Scanner

FILE_MANAGER

public static final FileManager FILE_MANAGER

Constant assigned with instance of #FileManager for Contract Manager's file operations

See Also:

#FileManager

REGEX

public static final java.lang.String REGEX

static Constant assigned for matching the company's new Reference Number Format Reference Number Format explained : -Two letter followed by three digits and a letter. -The last letter can only be B for Business Account, C for Charity Account or N for Non-Business Account "[A-Z]{2}\\d{3}[B|C|N]"

See Also:

Constant Field Values

PACKAGES

public static final java.util.Map<java.lang.Integer,java.lang.String> PACKAGES

static Constant assigned for the company offering packages Company providing packages : -Small 300 mins -Medium 600 mins -Large 1200 mins

DATA_BUNDLES

```
public static final java.util.Map<java.lang.Integer,java.lang.String> DATA_BUNDLES
```

static Constant assigned for the Data Bundles of the company Data Bundles of the company : -Low (1GB) -Medium (4GB) -High (8GB) -Unlimited

PERIOD_MAP

```
public static final java.util.Map<java.lang.Integer,java.lang.String> PERIOD_MAP
```

static Constant assigned for Contract periods as per the policies of the company Contract periods as per the policies of the company : 1, 12, 18 or 24 months

MONTH_MAP

```
public static final java.util.Map<java.lang.Integer,java.lang.String> MONTH_MAP
```

static Constant assigned to calculate the total number of contracts per month, assuming the files only include data for the current year only

Constructor Detail

InitContractManager

```
public InitContractManager()
```

Hierarchy For All Packages

Package Hierarchies:
contractmanager, contractmanager.maincode, contractmanager.model, contractmanager.util

Class Hierarchy

- java.lang.Object
  - contractmanager.model.**ArchiveModel** (implements java.io.Serializable)
  - contractmanager.**ContractManager**
  - contractmanager.util.**FileManager** (implements contractmanager.util.ArchiveWrapper, contractmanager.util.ContractWrapper)
  - contractmanager.maincode.**InitContractManager**
    - contractmanager.maincode.**ContractSummary**
    - contractmanager.maincode.**ContractUpdate**
    - contractmanager.maincode.**CreateContract**
    - contractmanager.maincode.**FindAndDisplayContract**

Interface Hierarchy

- contractmanager.util.**ArchiveWrapper**
- contractmanager.util.**ContractWrapper**

A

ArchiveModel - Class in contractmanager.model

A Class Model/Bean for initializing Contract Manager.

ArchiveModel() - Constructor for class contractmanager.model.ArchiveModel

ArchiveWrapper - Interface in contractmanager.util

The ArchiveWrapper is an Interface for Archive Operations Archive Operations specifically includes file operations regarding archive entries for Contract Manager

ContactManagerDocumentation

Packages

Package	Description
contractmanager	
contractmanager.maincode	
contractmanager.model	
contractmanager.util	

Package contractmanager

Class Summary

Class	Description
ContractManager	The Main Class for Contract Manager implements an Application/Prototype the Telephone Contract Manager.

Package contractmanager.maincode

Class Summary

Class	Description
ContractSummary	The Class ContractSummary provides the user about Summary of Contracts in our repositories ContractSummary to retrieve the - Total number of contracts - Number of contracts with High Data Bundle - Number of contracts with Unlimited Data Bundle - Average charge of large package contracts - Total number of contracts per month (assuming the files only include data for the current year only) Also show an entry for all twelve months even those that have an entry of zero
ContractUpdate	The Class ContractUpdate provides the client contract reference and their whole details user gets to search and fetch the contract details and then if needed, user can update the contract details but only if that contract meets the company's predetermined criteria user can search for the contract details and fetch the details
CreateContract	Class CreateContract for initializing new Contracts for the Contract Manager
FindAndDisplayContract	Class FindAndDisplayContract enables users to list out the contract data for the Contract Manager
InitContractManager	An Abstract Class for Contract Manager with essential Constants and Fields Initializing/Assigning Constant Fields : REGEX, PACKAGES, DATA_BUNDLES, PERIOD_MAP, MONTH_MAP to : -Create a Contract -Find And Display Contract(s) -Get Summary of a Contract(s) -Modify Existing Contract(s)

Package contractmanager.model

Class Summary

Class	Description
ArchiveModel	A Class Model/Bean for initializing Contract Manager.

Package contractmanager.util

Interface Summary

Interface	Description
ArchiveWrapper	The ArchiveWrapper is an Interface for Archive Operations Archive Operations specifically includes file operations regarding archive entries for Contract Manager
ContractWrapper	The ContractWrapper is an Interface for Contract Operations, specifically file operations regarding contract entries for Contract Manager

Class Summary

Class	Description
FileManager	Class FileManager is for Contract Manager's File operations

Uses of Class contractmanager.model.ArchiveModel

Packages that use ArchiveModel

Package	Description
contractmanager.util	

Uses of ArchiveModel in contractmanager.util

Methods in contractmanager.util that return types with arguments of type ArchiveModel

Modifier and Type	Method and Description
`java.util.LinkedHashMap<java.lang.Integer,ArchiveModel>`	**FileManager.readArchiveFromFile()** read Contract details to Archive repository(File) ArchiveWrapper implementation in FileManager
`java.util.LinkedHashMap<java.lang.Integer,ArchiveModel>`	**ArchiveWrapper.readArchiveFromFile()** ArchiveWrapper implementation in FileManager

Hierarchy For Package contractmanager.model

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - contractmanager.model.**ArchiveModel** (implements java.io.Serializable)

All Classes

Serialized Form

Package contractmanager.model

Class contractmanager.model.ArchiveModel extends java.lang.Object implements Serializable

Serialized Fields

totalNo

```
long totalNo
```

totalHighOrUnlimited

```
long totalHighOrUnlimited
```

totalLargePackageCharge

```
double totalLargePackageCharge
```

totalLargePackage

```
long totalLargePackage
```

Uses of Interface contractmanager.util.ArchiveWrapper

Packages that use ArchiveWrapper

Package	Description
contractmanager.util	

Uses of ArchiveWrapper in contractmanager.util

Classes in contractmanager.util that implement ArchiveWrapper

Modifier and Type	Class and Description
class	**FileManager** Class FileManager is for Contract Manager's File operations

Hierarchy For Package contractmanager.util

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - contractmanager.util.**FileManager** (implements contractmanager.util.ArchiveWrapper, contractmanager.util.ContractWrapper)

Interface Hierarchy

- contractmanager.util.**ArchiveWrapper**
- contractmanager.util.**ContractWrapper**

Hierarchy For Package contractmanager

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - contractmanager.**ContractManager**

Hierarchy For Package contractmanager.maincode

Package Hierarchies:
All Packages

Class Hierarchy

- java.lang.Object
  - contractmanager.maincode.**InitContractManager**
    - contractmanager.maincode.**ContractSummary**
    - contractmanager.maincode.**ContractUpdate**
    - contractmanager.maincode.**CreateContract**
    - contractmanager.maincode.**FindAndDisplayContract**

C

D

F

G

I

M

P

PACKAGES - Static variable in class contractmanager.maincode.InitContractManager

static Constant assigned for the company offering packages Company providing packages : -Small 300 mins -Medium 600 mins -Large 1200 mins

PERIOD_MAP - Static variable in class contractmanager.maincode.InitContractManager

static Constant assigned for Contract periods as per the policies of the company Contract periods as per the policies of the company : 1, 12, 18 or 24 months

R

S

U

updateExisitngContract() - Method in class contractmanager.maincode.ContractUpdate

To update any contract user has to find the contract first, henceforth the user is asked for a Search keyword to find the contract that he intends to update.

W

Uses of Package contractmanager.maincode

Packages that use contractmanager.maincode

Package	Description
contractmanager.maincode	

Classes in contractmanager.maincode used by contractmanager.maincode

Class and Description
InitContractManager An Abstract Class for Contract Manager with essential Constants and Fields Initializing/Assigning Constant Fields : REGEX, PACKAGES, DATA_BUNDLES, PERIOD_MAP, MONTH_MAP to : -Create a Contract -Find And Display Contract(s) -Get Summary of a Contract(s) -Modify Existing Contract(s)

Uses of Package contractmanager.model

Packages that use contractmanager.model

Package	Description
contractmanager.util	

Classes in contractmanager.model used by contractmanager.util

Class and Description
ArchiveModel A Class Model/Bean for initializing Contract Manager.

Uses of Package contractmanager.util

Packages that use contractmanager.util

Package	Description
contractmanager.maincode	
contractmanager.util	

Classes in contractmanager.util used by contractmanager.maincode

Class and Description
FileManager Class FileManager is for Contract Manager's File operations

Classes in contractmanager.util used by contractmanager.util

Class and Description
ArchiveWrapper The ArchiveWrapper is an Interface for Archive Operations Archive Operations specifically includes file operations regarding archive entries for Contract Manager
ContractWrapper The ContractWrapper is an Interface for Contract Operations, specifically file operations regarding contract entries for Contract Manager