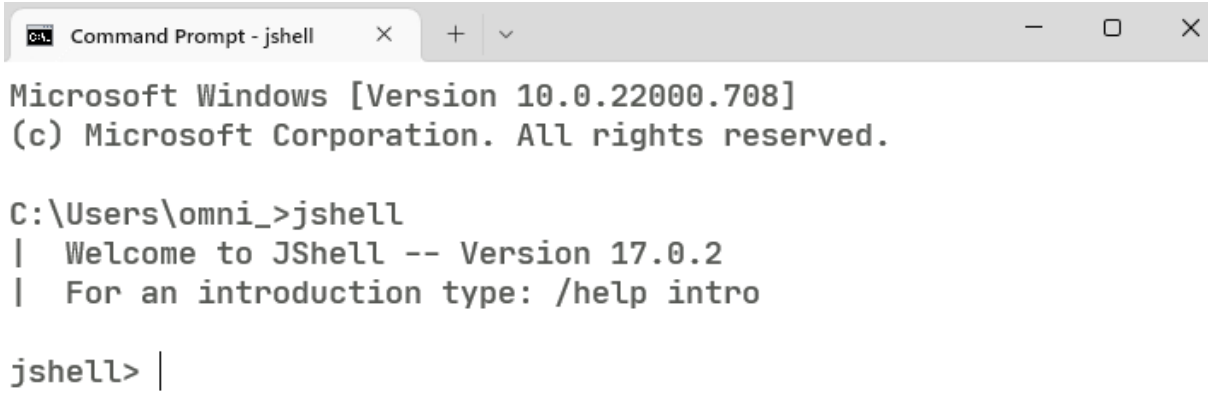
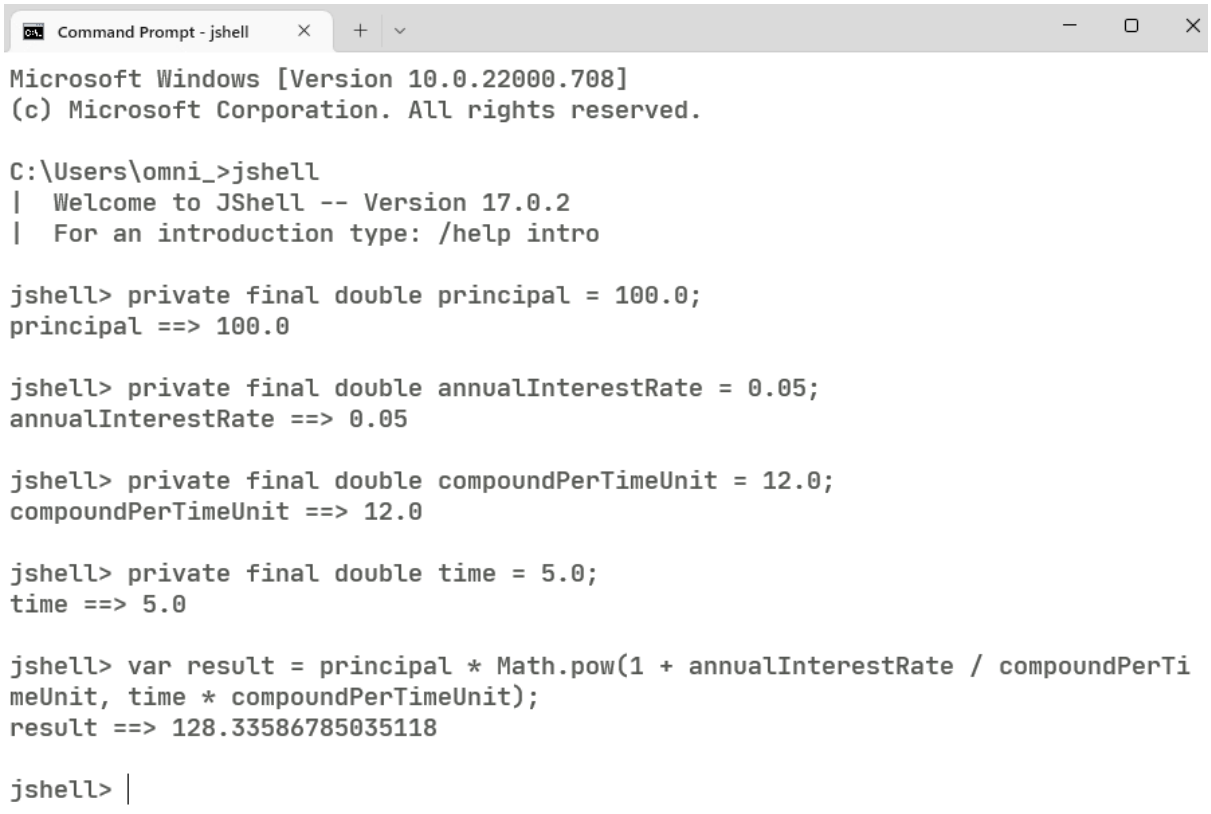
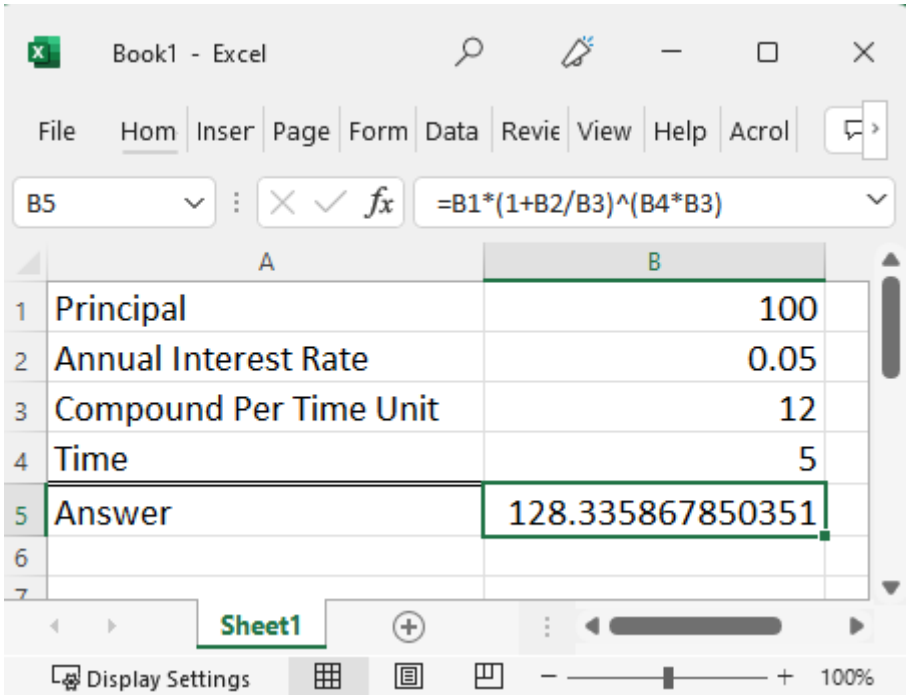
Chapter 1: Understanding Java Distributions

No images…

Chapter 2: Code, Compile, and Execute






Book1 - Excel
File
Hom
Inser
Page
Form
Data
Revie
View
Help
Acrol
B5
=B1*(1+B2/B3)^(B4*B3)
A
B
1 Principal 100
2 Annual Interest Rate 0.05
3 Compound Per Time Unit 12
4 Time 5
5 Answer 128.335867850351
6
Sheet1
Display Settings
100%

JShell Edit Pad

```
import java.text.NumberFormat;

public class CompoundInterest01 {

    private final double principal = 100.0;
    private final double annualInterestRate = 0.05;
    private final double compoundPerTimeUnit = 12.0;
    private final double time = 5.0; //

    private final NumberFormat currencyFormat;
    private final NumberFormat percentFormat;

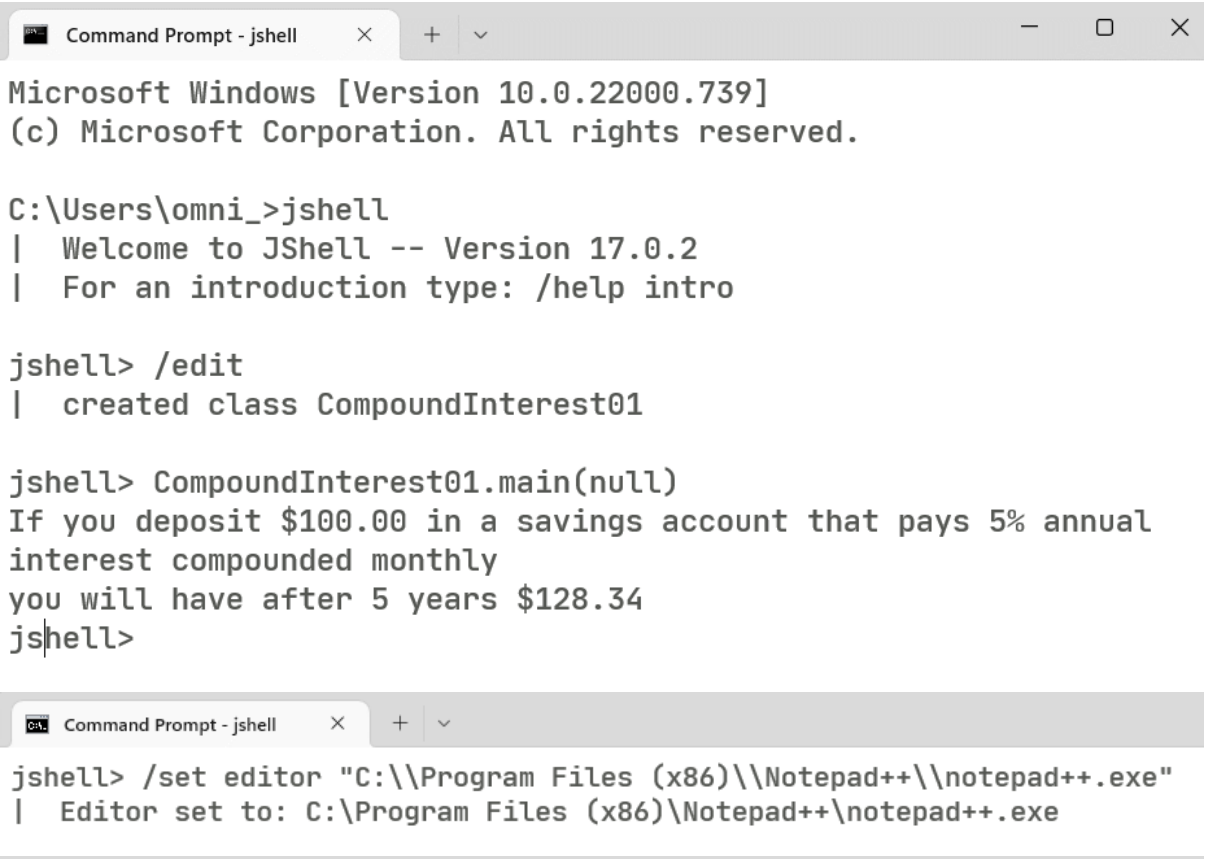
    public CompoundInterest01() {
        currencyFormat = NumberFormat.getCurrencyInstance();
        percentFormat = NumberFormat.getPercentInstance();
        percentFormat.setMinimumFractionDigits(0);
        percentFormat.setMaximumFractionDigits(5);
    }

    public void perform() {
        var result = calculateCompoundInterest();
        System.out.printf("If you deposit %s in a savings account "
                + "that pays %s annual interest compounded monthly%n"
                + "you will have after %1.0f years %s",
                currencyFormat.format(principal),
                percentFormat.format(annualInterestRate),
                time, currencyFormat.format(result));
    }

    private double calculateCompoundInterest() {
        var result = principal * Math.pow(1 + annualInterestRate /
                compoundPerTimeUnit, time * compoundPerTimeUnit);
        return result;
    }

    public static void main(String[] args) {
        var banker = new CompoundInterest01();
        banker.perform();
    }
}
```

Cancel Accept Exit


Command Prompt - jshell
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.
C:\Users\omni_>jshell
| Welcome to JShell -- Version 17.0.2
| For an introduction type: /help intro
jshell> /edit
| created class CompoundInterest01
jshell> CompoundInterest01.main(null)
If you deposit $100.00 in a savings account that pays 5% annual
interest compounded monthly
you will have after 5 years $128.34
jshell>
Command Prompt - jshell
jshell> /set editor "C:\\Program Files (x86)\\Notepad++\\notepad++.exe"
| Editor set to: C:\Program Files (x86)\Notepad++\notepad++.exe
Command Prompt
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.
C:\PacktJavaCode\CompoundInterest01>javac CompoundInterest01.java
C:\PacktJavaCode\CompoundInterest01>dir
Volume in drive C has no label.
Volume Serial Number is 46B1-7989
Directory of C:\PacktJavaCode\CompoundInterest01
2022-06-15 03:03 PM <DIR> .
2022-06-15 02:57 PM <DIR> ..
2022-06-15 04:02 PM 1,636 CompoundInterest01.class
2022-06-15 04:01 PM 1,433 CompoundInterest01.java
2 File(s) 3,069 bytes
2 Dir(s) 672,338,796,544 bytes free
C:\PacktJavaCode\CompoundInterest01>

```
Command Prompt
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\PacktJavaCode\CompoundInterest01>javac CompoundInterest01.java

C:\PacktJavaCode\CompoundInterest01>dir
 Volume in drive C has no label.
 Volume Serial Number is 46B1-7989

 Directory of C:\PacktJavaCode\CompoundInterest01

2022-06-15  03:03 PM    <DIR>          .
2022-06-15  02:57 PM    <DIR>          ..
2022-06-15  04:02 PM             1,636 CompoundInterest01.class
2022-06-15  04:01 PM             1,433 CompoundInterest01.java
               2 File(s)          3,069 bytes
               2 Dir(s)  672,338,796,544 bytes free

C:\PacktJavaCode\CompoundInterest01>java CompoundInterest01
If you deposit $100.00 in a savings account that pays 5% annual in
terest compounded monthly
you will have after 5 years $128.34
C:\PacktJavaCode\CompoundInterest01>
```

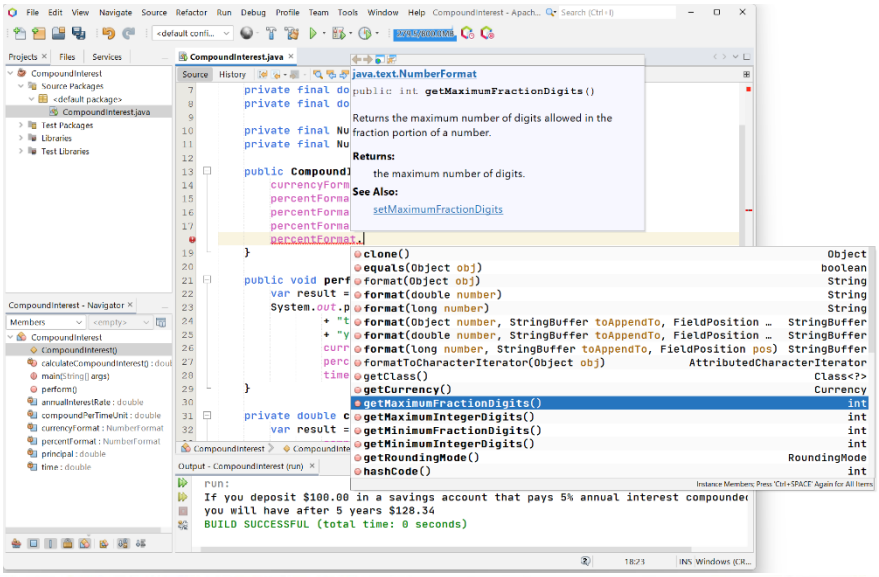
```
Command Prompt
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

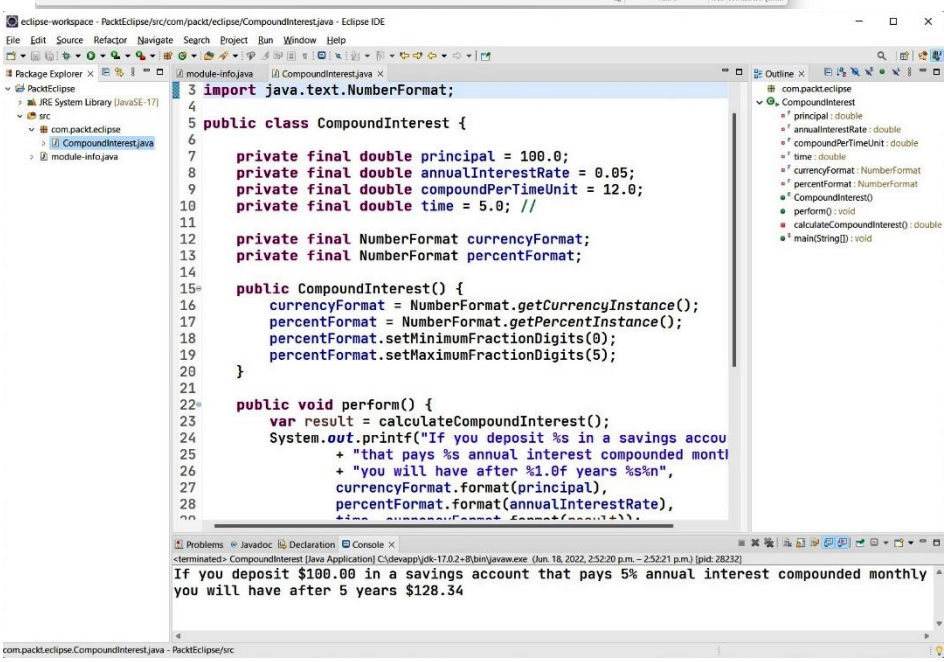
C:\PacktJavaCode\CompoundInterest01>java CompoundInterest01.java
If you deposit $100.00 in a savings account that pays 5% annual interest
compounded monthly
you will have after 5 years $128.34
C:\PacktJavaCode\CompoundInterest01>
```

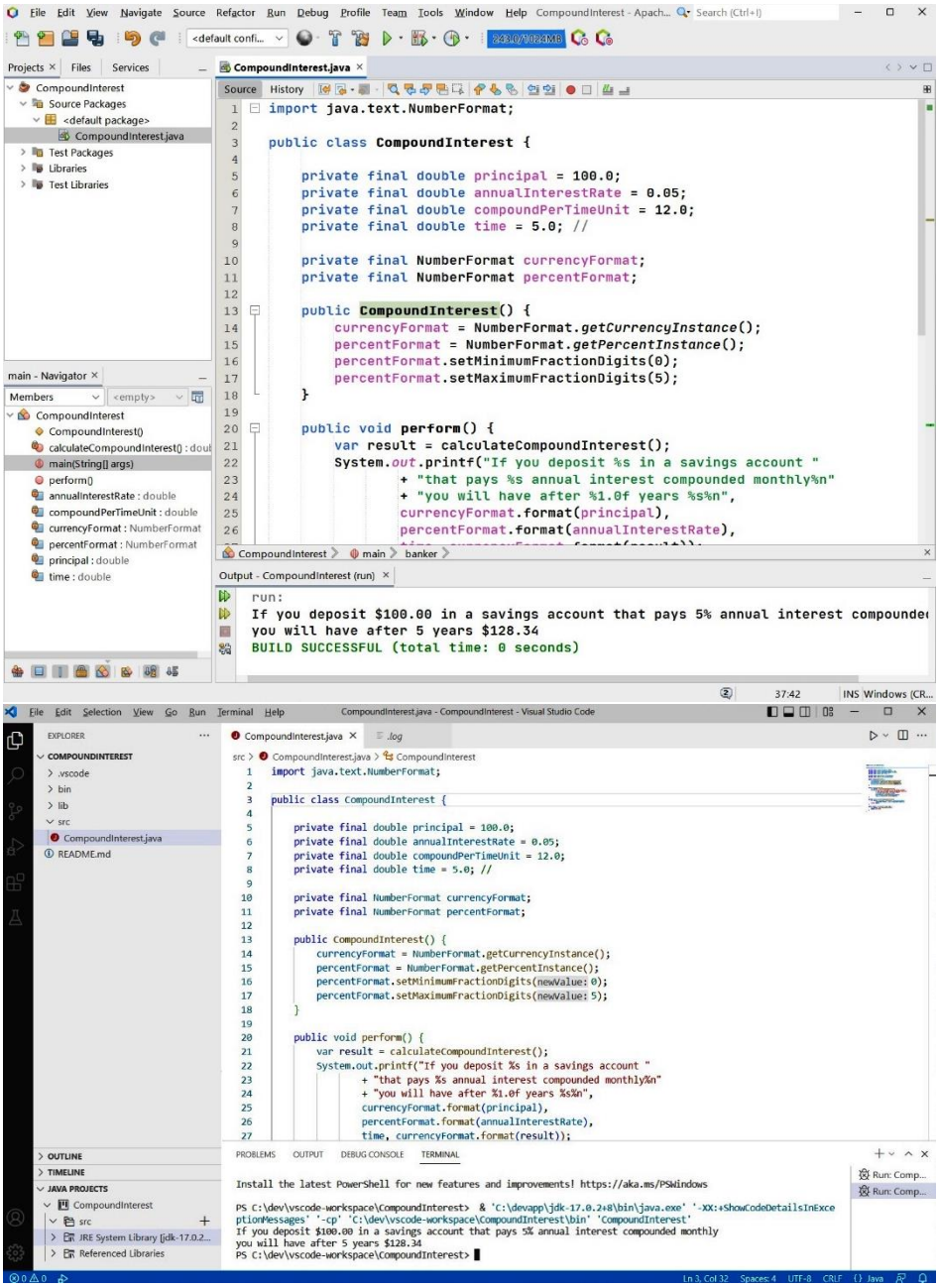
```
Command Prompt
Microsoft Windows [Version 10.0.22000.739]
(c) Microsoft Corporation. All rights reserved.

C:\PacktJavaCode\CompoundInterest02>java CompoundInterest02.java
If you deposit $100.00 in a savings account that pays 5% annual interest
compounded monthly
you will have after 5 years $128.34
C:\PacktJavaCode\CompoundInterest02>
```

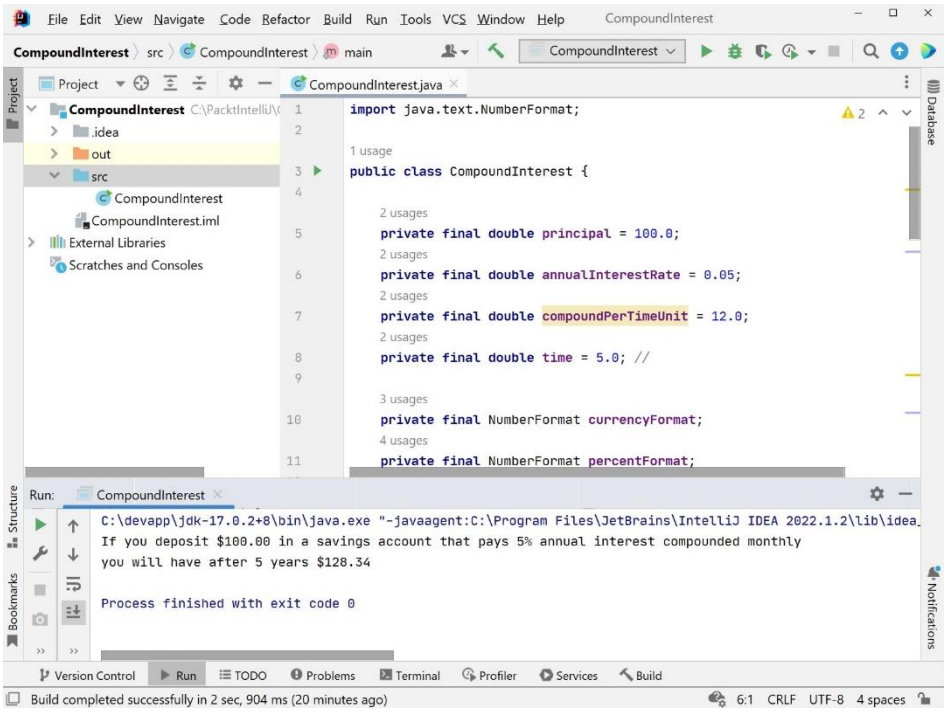
```
omniprof@Kong:~/CompoundInterest03$ ./CompoundInterest03
If you deposit $100.00 in a savings account that pays 5% annual interest compounded monthly
you will have after 5 years $128.34
omniprof@Kong:~/CompoundInterest03$
```


java.text.NumberFormat
public int getMaximumFractionDigits()
Returns the maximum number of digits allowed in the fraction portion of a number.
Returns:
the maximum number of digits.
See Also:
setMaximumFractionDigits
clone() Object
equals(Object obj) boolean
format(Object obj) String
format(double number) String
format(long number) String
format(Object number, StringBuffer toAppendTo, FieldPosition … StringBuffer
format(double number, StringBuffer toAppendTo, FieldPosition … StringBuffer
format(long number, StringBuffer toAppendTo, FieldPosition pos) StringBuffer
formatToCharacterIterator(Object obj) AttributedCharacterIterator
getClass() Class<?>
getCurrency() Currency
getMaximumFractionDigits() int
getMaximumIntegerDigits() int
getMinimumFractionDigits() int
getMinimumIntegerDigits() int
getRoundingMode() RoundingMode
hashCode() int
run:
If you deposit $100.00 in a savings account that pays 5% annual interest compounde
you will have after 5 years $128.34
BUILD SUCCESSFUL (total time: 0 seconds)


import java.text.NumberFormat;
public class CompoundInterest {
private final double principal = 100.0;
private final double annualInterestRate = 0.05;
private final double compoundPerTimeUnit = 12.0;
private final double time = 5.0; //
private final NumberFormat currencyFormat;
private final NumberFormat percentFormat;
public CompoundInterest() {
currencyFormat = NumberFormat.getCurrencyInstance();
percentFormat = NumberFormat.getPercentInstance();
percentFormat.setMinimumFractionDigits(0);
percentFormat.setMaximumFractionDigits(5);
}
public void perform() {
var result = calculateCompoundInterest();
System.out.printf("If you deposit %s in a savings accou
+ "that pays %s annual interest compounded monthl
+ "you will have after %1.0f years %s%n",
currencyFormat.format(principal),
percentFormat.format(annualInterestRate),
If you deposit $100.00 in a savings account that pays 5% annual interest compounded monthly
you will have after 5 years $128.34


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help CompoundInterest - Apach... Search (Ctrl+I)
Projects Files Services
CompoundInterest
Source Packages
<default package>
CompoundInterest.java
Test Packages
Libraries
Test Libraries
CompoundInterest.java
Source History
import java.text.NumberFormat;
public class CompoundInterest {
private final double principal = 100.0;
private final double annualInterestRate = 0.05;
private final double compoundPerTimeUnit = 12.0;
private final double time = 5.0; //
private final NumberFormat currencyFormat;
private final NumberFormat percentFormat;
public CompoundInterest() {
currencyFormat = NumberFormat.getCurrencyInstance();
percentFormat = NumberFormat.getPercentInstance();
percentFormat.setMinimumFractionDigits(0);
percentFormat.setMaximumFractionDigits(5);
}
public void perform() {
var result = calculateCompoundInterest();
System.out.printf("If you deposit %s in a savings account "
+ "that pays %s annual interest compounded monthly%n"
+ "you will have after %1.0f years %s%n",
currencyFormat.format(principal),
percentFormat.format(annualInterestRate),
main - Navigator
Members
CompoundInterest
CompoundInterest()
calculateCompoundInterest() : dou
main(String[] args)
perform()
annualInterestRate : double
compoundPerTimeUnit : double
currencyFormat : NumberFormat
percentFormat : NumberFormat
principal : double
time : double
CompoundInterest main banker
Output - CompoundInterest (run)
run:
If you deposit $100.00 in a savings account that pays 5% annual interest compounde
you will have after 5 years $128.34
BUILD SUCCESSFUL (total time: 0 seconds)
37:42
INS Windows (CR...

File Edit Selection View Go Run Terminal Help CompoundInterest.java - CompoundInterest - Visual Studio Code
EXPLORER
COMPOUNDINTEREST
.vscode
bin
lib
src
CompoundInterest.java
README.md
CompoundInterest.java .log
src > CompoundInterest.java > CompoundInterest
import java.text.NumberFormat;
public class CompoundInterest {
private final double principal = 100.0;
private final double annualInterestRate = 0.05;
private final double compoundPerTimeUnit = 12.0;
private final double time = 5.0; //
private final NumberFormat currencyFormat;
private final NumberFormat percentFormat;
public CompoundInterest() {
currencyFormat = NumberFormat.getCurrencyInstance();
percentFormat = NumberFormat.getPercentInstance();
percentFormat.setMinimumFractionDigits(newValue: 0);
percentFormat.setMaximumFractionDigits(newValue: 5);
}
public void perform() {
var result = calculateCompoundInterest();
System.out.printf("If you deposit %s in a savings account "
+ "that pays %s annual interest compounded monthly%n"
+ "you will have after %1.0f years %s%n",
currencyFormat.format(principal),
percentFormat.format(annualInterestRate),
time, currencyFormat.format(result));
OUTLINE
TIMELINE
JAVA PROJECTS
CompoundInterest
src
JRE System Library [jdk-17.0.2...
Referenced Libraries
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
PS C:\dev\vscode-workspace\CompoundInterest> & 'C:\devapp\jdk-17.0.2+8\bin\java.exe' '-XX:+ShowCodeDetailsInExce
ptionMessages' '-cp' 'C:\dev\vscode-workspace\CompoundInterest\bin' 'CompoundInterest'
If you deposit $100.00 in a savings account that pays 5% annual interest compounded monthly
you will have after 5 years $128.34
PS C:\dev\vscode-workspace\CompoundInterest>
Run: Comp...
Run: Comp...
Ln 3, Col 32 Spaces: 4 UTF-8 CRLF Java

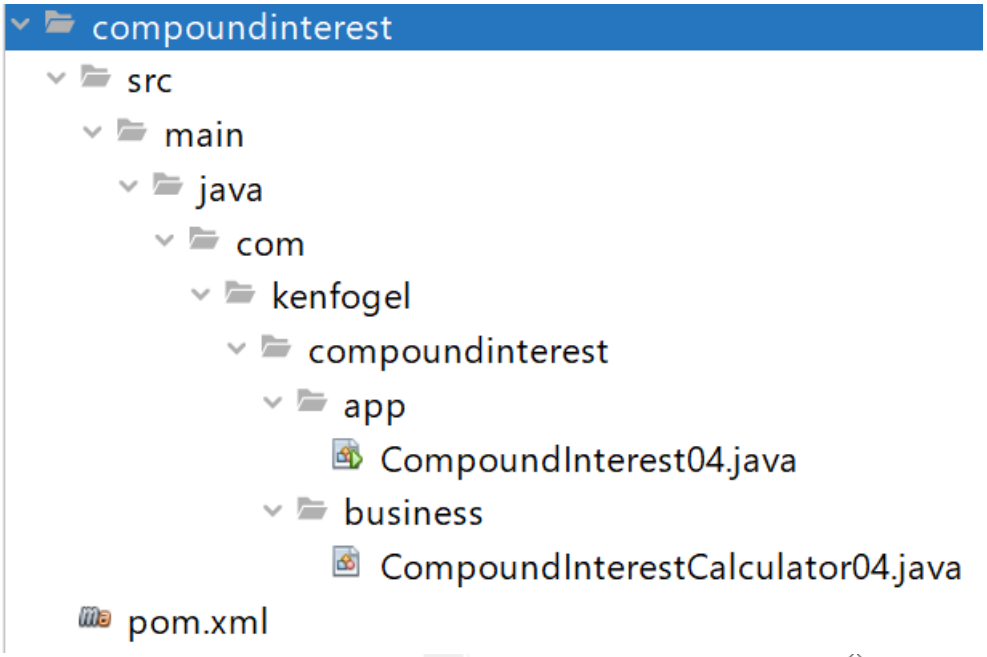

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
CompoundInterest
CompoundInterest src CompoundInterest main
CompoundInterest
Project
CompoundInterest.java
CompoundInterest C:\PacktIntelliJ\
.idea
out
src
CompoundInterest
CompoundInterest.iml
External Libraries
Scratches and Consoles
import java.text.NumberFormat;
1 usage
public class CompoundInterest {
2 usages
private final double principal = 100.0;
2 usages
private final double annualInterestRate = 0.05;
2 usages
private final double compoundPerTimeUnit = 12.0;
2 usages
private final double time = 5.0; //
3 usages
private final NumberFormat currencyFormat;
4 usages
private final NumberFormat percentFormat;
Run:
CompoundInterest
C:\devapp\jdk-17.0.2+8\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.1.2\lib\idea_
If you deposit $100.00 in a savings account that pays 5% annual interest compounded monthly
you will have after 5 years $128.34
Process finished with exit code 0
Version Control
Run
TODO
Problems
Terminal
Profiler
Services
Build
Build completed successfully in 2 sec, 904 ms (20 minutes ago)
6:1 CRLF UTF-8 4 spaces

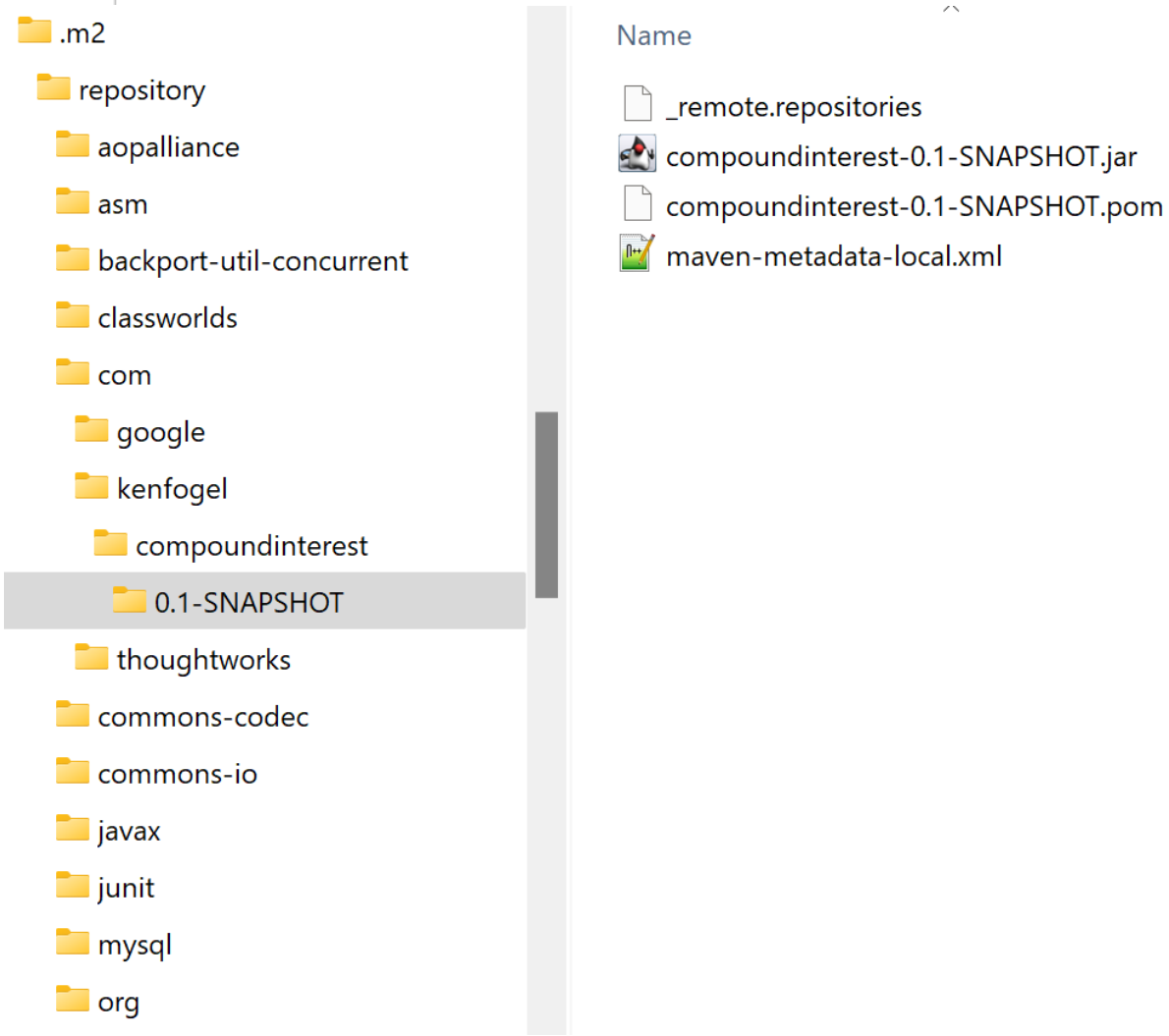
Chapter 3: The Maven Build Tool

Binary tar.gz archive	apache-maven-3.8.6-bin.tar.gz
Binary zip archive	apache-maven-3.8.6-bin.zip

```
C:\PacktJavaCode\CompoundInterest04>mvn --version
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\devapp\apache-maven-3.8.6
Java version: 17.0.2, vendor: Eclipse Adoptium, runtime: C:\devapp\jdk-17.0.2+8
Default locale: en_CA, platform encoding: Cp1252
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
```

```
omniprof@Kong:~$ mvn --version
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: /home/omniprof/apache-maven-3.8.6
Java version: 17.0.3, vendor: Eclipse Adoptium, runtime: /home/omniprof/jdk-17.0.3+7
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.10.102.1-microsoft-standard-wsl2", arch: "amd64", family: "unix"
```


compoundinterest
src
main
java
com
kenfogel
compoundinterest
app
CompoundInterest04.java
business
CompoundInterestCalculator04.java
pom.xml


.m2
repository
aopalliance
asm
backport-util-concurrent
classworlds
com
google
kenfogel
compoundinterest
0.1-SNAPSHOT
thoughtworks
commons-codec
commons-io
javax
junit
mysql
org
Name
_remote.repositories
compoundinterest-0.1-SNAPSHOT.jar
compoundinterest-0.1-SNAPSHOT.pom
maven-metadata-local.xml

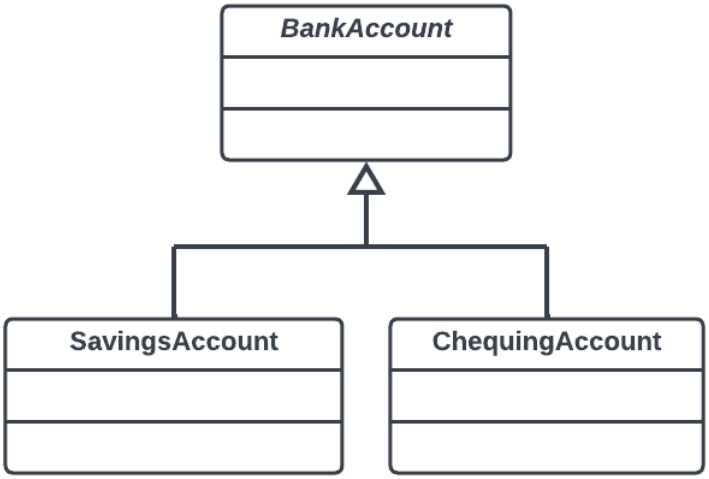
Chapter 4: Language Fundamentals – Data Types and Variables

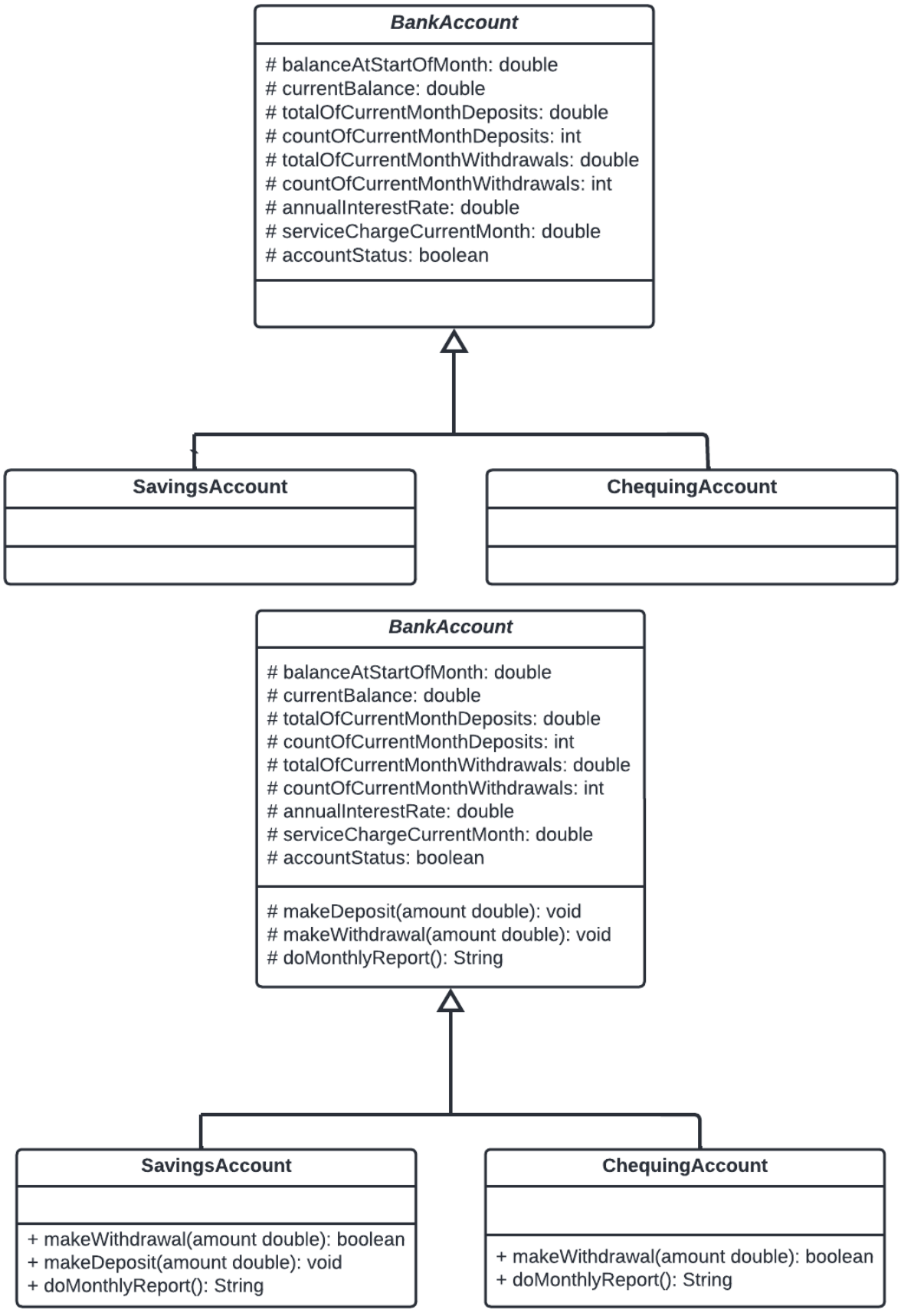
No images…

Chapter 5: Language Fundamentals – Classes

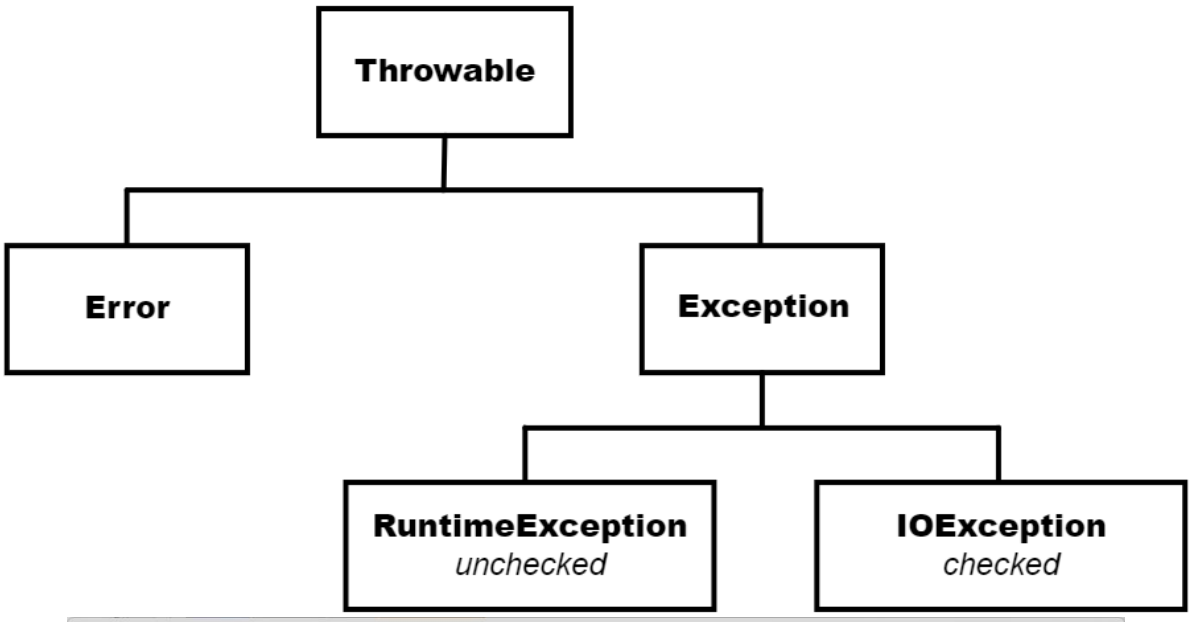
No images…

Chapter 6: Methods, Interfaces, Records, and Their Relationships




BankAccount
balanceAtStartOfMonth: double
currentBalance: double
totalOfCurrentMonthDeposits: double
countOfCurrentMonthDeposits: int
totalOfCurrentMonthWithdrawals: double
countOfCurrentMonthWithdrawals: int
annualInterestRate: double
serviceChargeCurrentMonth: double
accountStatus: boolean
SavingsAccount
ChequingAccount
BankAccount
balanceAtStartOfMonth: double
currentBalance: double
totalOfCurrentMonthDeposits: double
countOfCurrentMonthDeposits: int
totalOfCurrentMonthWithdrawals: double
countOfCurrentMonthWithdrawals: int
annualInterestRate: double
serviceChargeCurrentMonth: double
accountStatus: boolean
makeDeposit(amount double): void
makeWithdrawal(amount double): void
doMonthlyReport(): String
SavingsAccount
+ makeWithdrawal(amount double): boolean
+ makeDeposit(amount double): void
+ doMonthlyReport(): String
ChequingAccount
+ makeWithdrawal(amount double): boolean
+ doMonthlyReport(): String

Chapter 7: Java Syntax and Exceptions



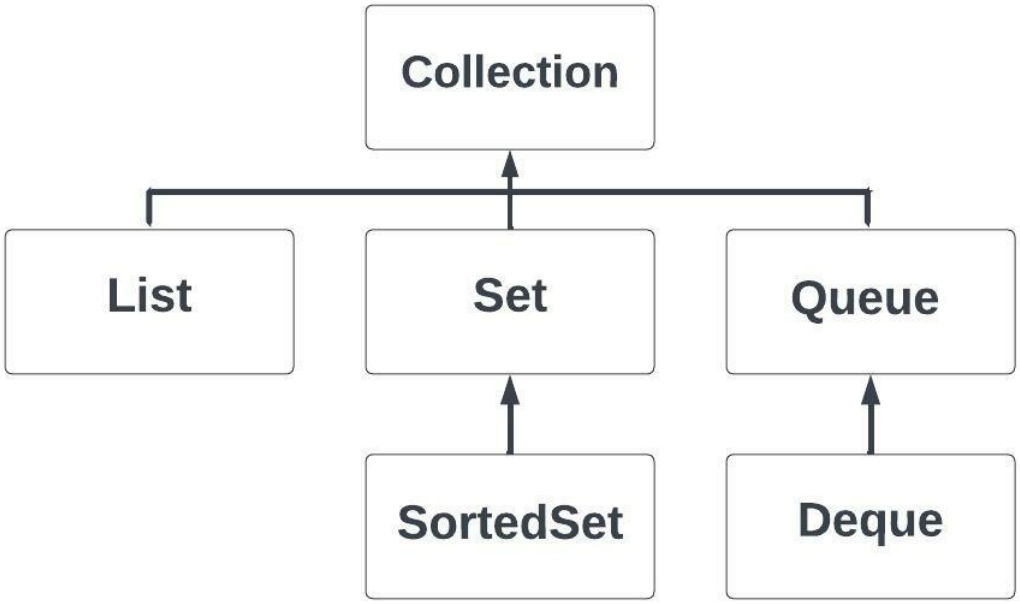

```
Command Prompt

C:\PacktJavaCode>java FileException.java
java.nio.file.NoSuchFileException: c:\temp\textile.txt
        at java.base/sun.nio.fs.WindowsException.translateToIOException(WindowsException.java:85)
        at java.base/sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:103)
        at java.base/sun.nio.fs.WindowsException.rethrowAsIOException(WindowsException.java:108)
        at java.base/sun.nio.fs.WindowsFileSystemProvider.newByteChannel(WindowsFileSystemProvider.java:236)
        at java.base/java.nio.file.Files.newByteChannel(Files.java:380)
        at java.base/java.nio.file.Files.newByteChannel(Files.java:432)
        at java.base/java.nio.file.spi.FileSystemProvider.newInputStream(FileSystemProvider.java:422)
        at java.base/java.nio.file.Files.newInputStream(Files.java:160)
        at java.base/java.nio.file.Files.newBufferedReader(Files.java:2921)
        at java.base/java.nio.file.Files.readAllLines(Files.java:3411)
        at java.base/java.nio.file.Files.readAllLines(Files.java:3452)
        at FileException.doCheckedException(FileException.java:17)
        at FileException.perform(FileException.java:63)
        at FileException.main(FileException.java:69)
        at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMethodHandleAccessor.java:104)
        at java.base/java.lang.reflect.Method.invoke(Method.java:578)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.execute(Main.java:434)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.run(Main.java:205)
        at jdk.compiler/com.sun.tools.javac.launcher.Main.main(Main.java:132)

C:\PacktJavaCode>
```

Chapter 8: Arrays, Collections, Generics, Functions, and Streams

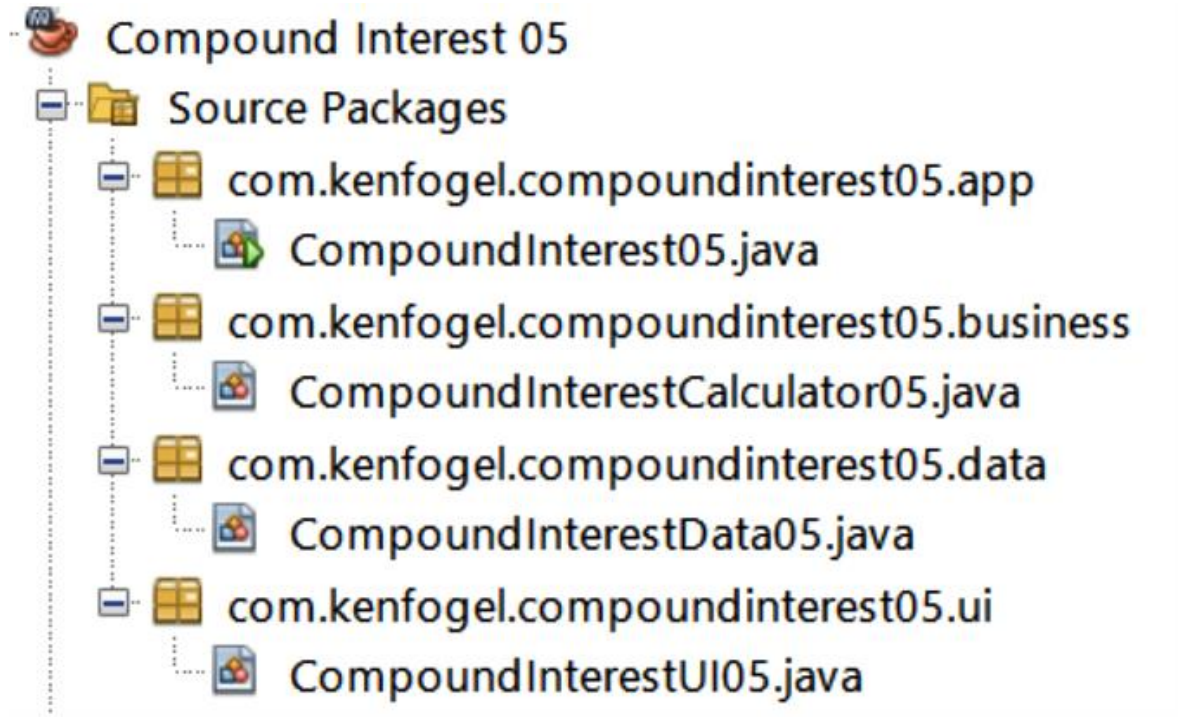




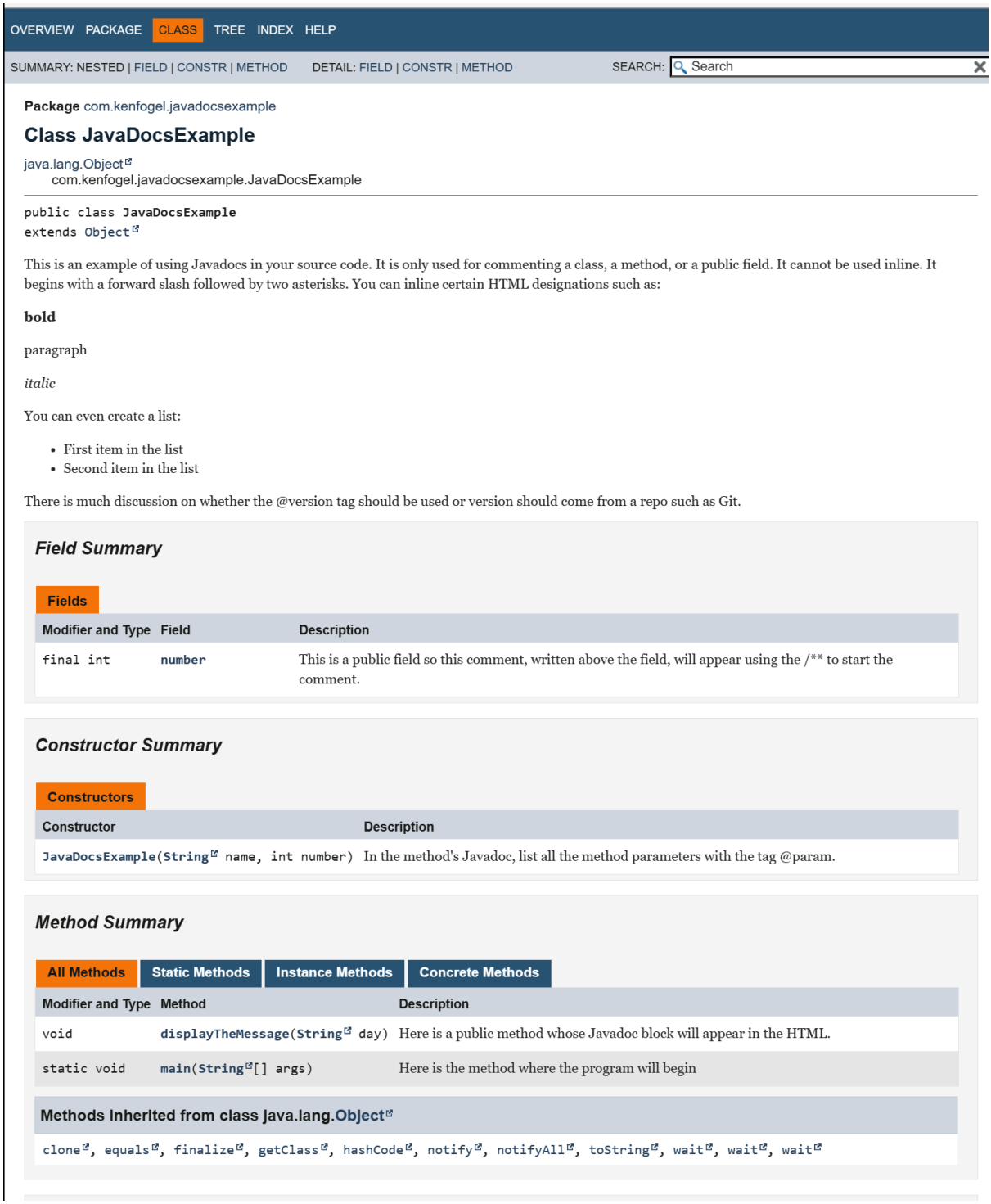
Chapter 9: Using Threads in Java

No images…

Chapter 10: Implementing Software Design Principles and Patterns in Java



Chapter 11: Documentation and Logging



OVERVIEW PACKAGE CLASS TREE INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH: Search

Package com.kenfogel.javadocsexample

Class JavaDocsExample

java.lang.Object
com.kenfogel.javadocsexample.JavaDocsExample

```
public class JavaDocsExample
extends Object
```

This is an example of using Javadocs in your source code. It is only used for commenting a class, a method, or a public field. It cannot be used inline. It begins with a forward slash followed by two asterisks. You can inline certain HTML designations such as:

bold

paragraph

italic

You can even create a list:

- First item in the list
- Second item in the list

There is much discussion on whether the @version tag should be used or version should come from a repo such as Git.

Field Summary

Fields

Modifier and Type	Field	Description
final int	number	This is a public field so this comment, written above the field, will appear using the /** to start the comment.

Constructor Summary

Constructors

Constructor	Description
JavaDocsExample(String name, int number)	In the method's Javadoc, list all the method parameters with the tag @param.

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
void	displayTheMessage(String day)	Here is a public method whose Javadoc block will appear in the HTML.
static void	main(String[] args)	Here is the method where the program will begin

Methods inherited from class java.lang.Object

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

Field Details

number

```
public final int number
```

This is a public field so this comment, written above the field, will appear using the /** to start the comment.

Constructor Details

JavaDocsExample

```
public JavaDocsExample(String name,
                       int number)
```

In the method's Javadoc, list all the method parameters with the tag @param. That this is also a constructor is recognized as the method has the same name as the class and does not have a return type.

Parameters:

`name` - user's name

`number` - answer to the ultimate question

Method Details

displayTheMessage

```
public void displayTheMessage(String day)
```

Here is a public method whose Javadoc block will appear in the HTML.

Parameters:

`day` - day of the week

main

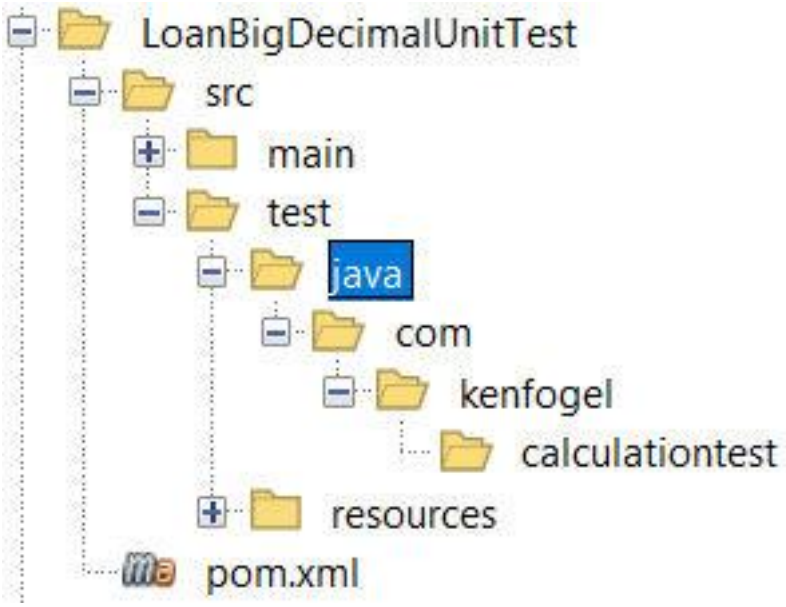
```
public static void main(String[] args)
```

Here is the method where the program will begin

Parameters:

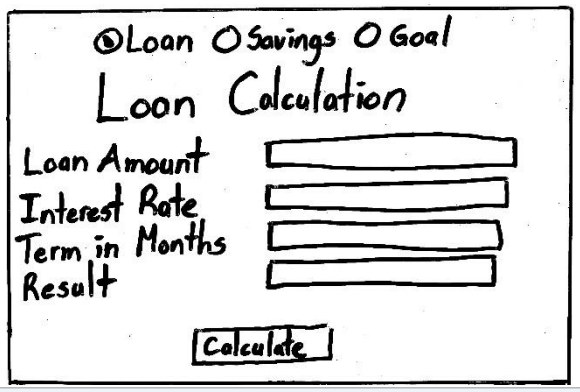
`args` - values passed at the command line

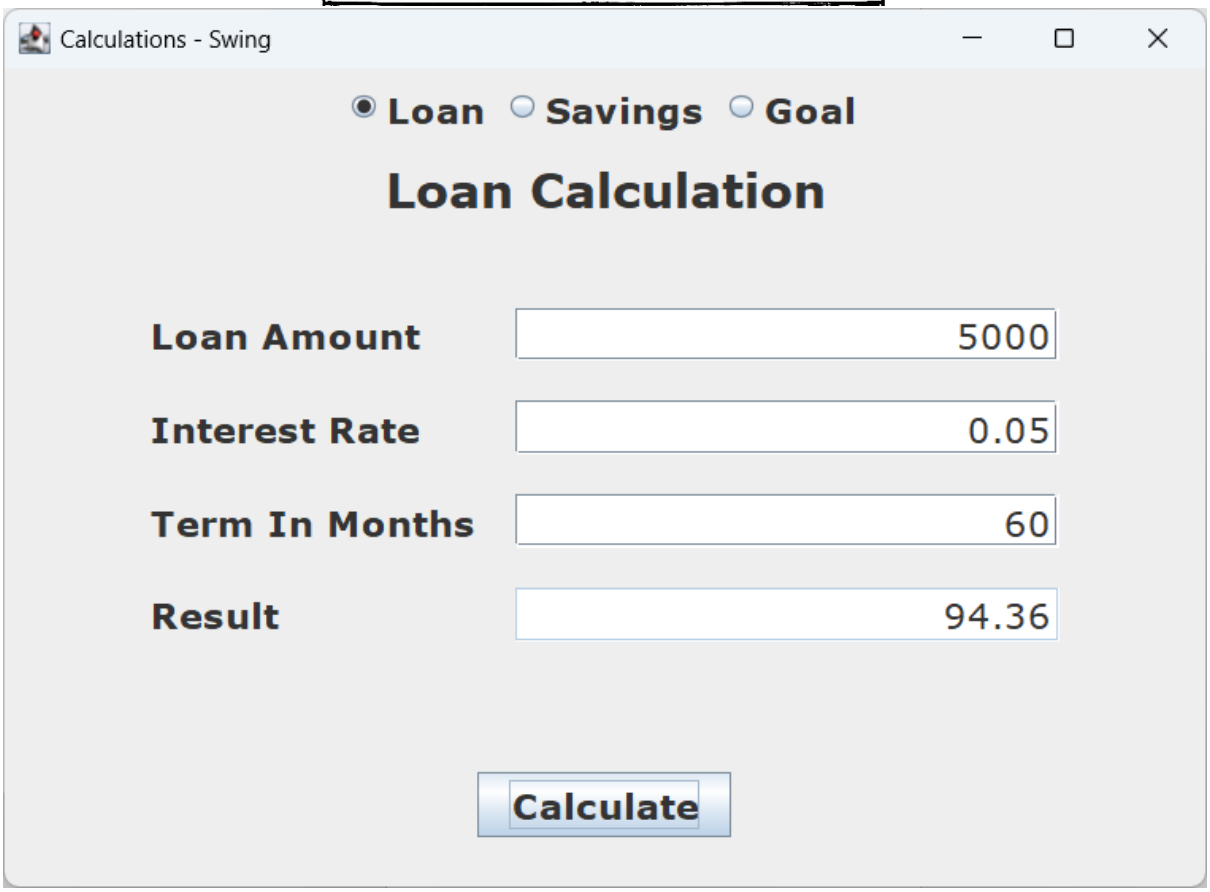
Chapter 12: BigDecimal and Unit Testing

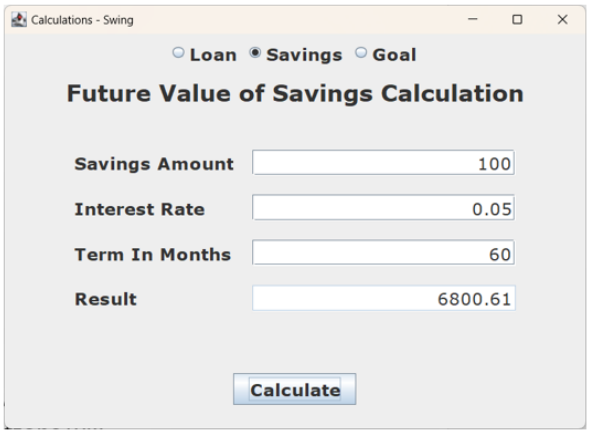


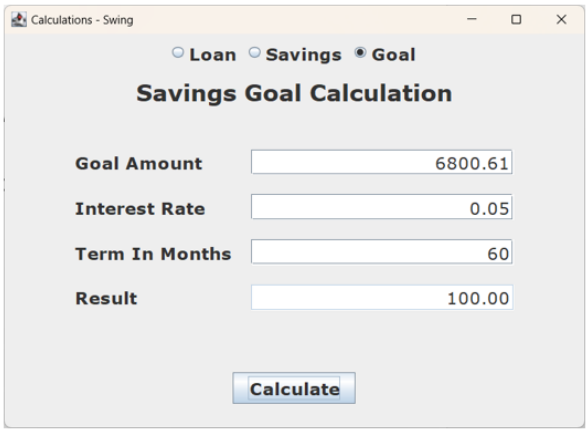

```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running com.kenfogel.calculationtest.SimpleTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.046 s -
in com.kenfogel.calculationtest.SimpleTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.497 s
[INFO] Finished at: 2022-12-21T19:10:09-05:00
[INFO] ------------------------------------------------------------------------
```

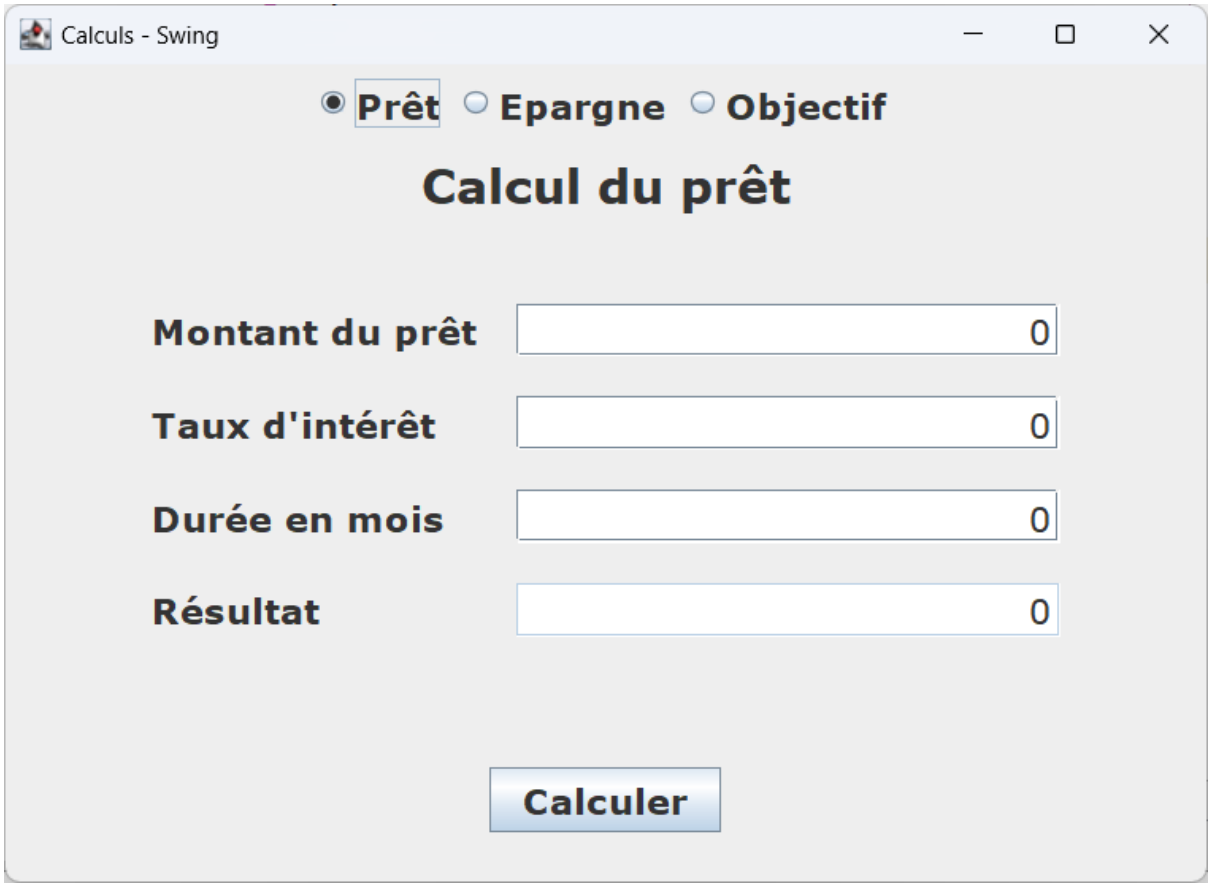
Chapter 13: Desktop Graphical User Interface Coding with Swing and JavaFX

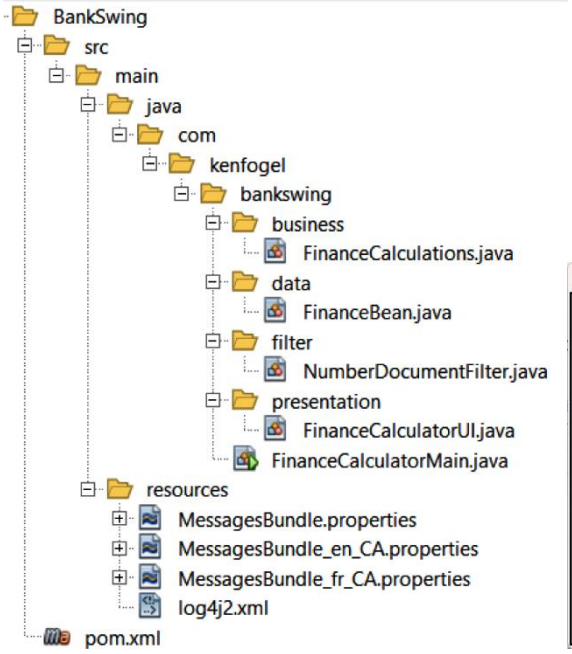


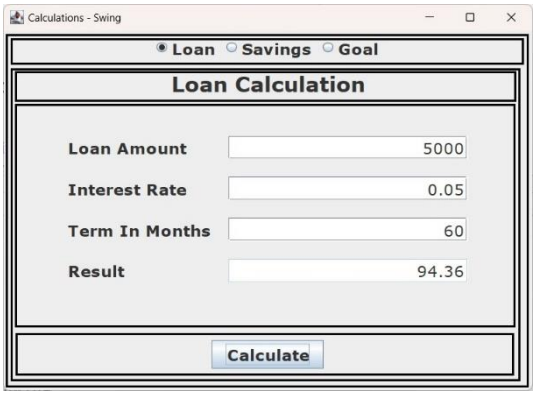


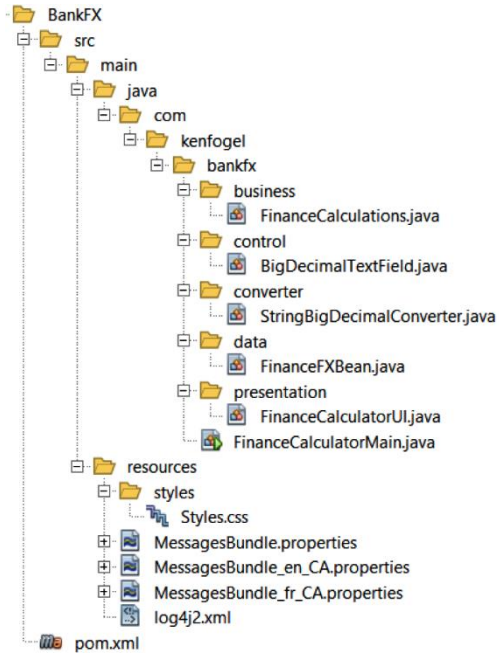




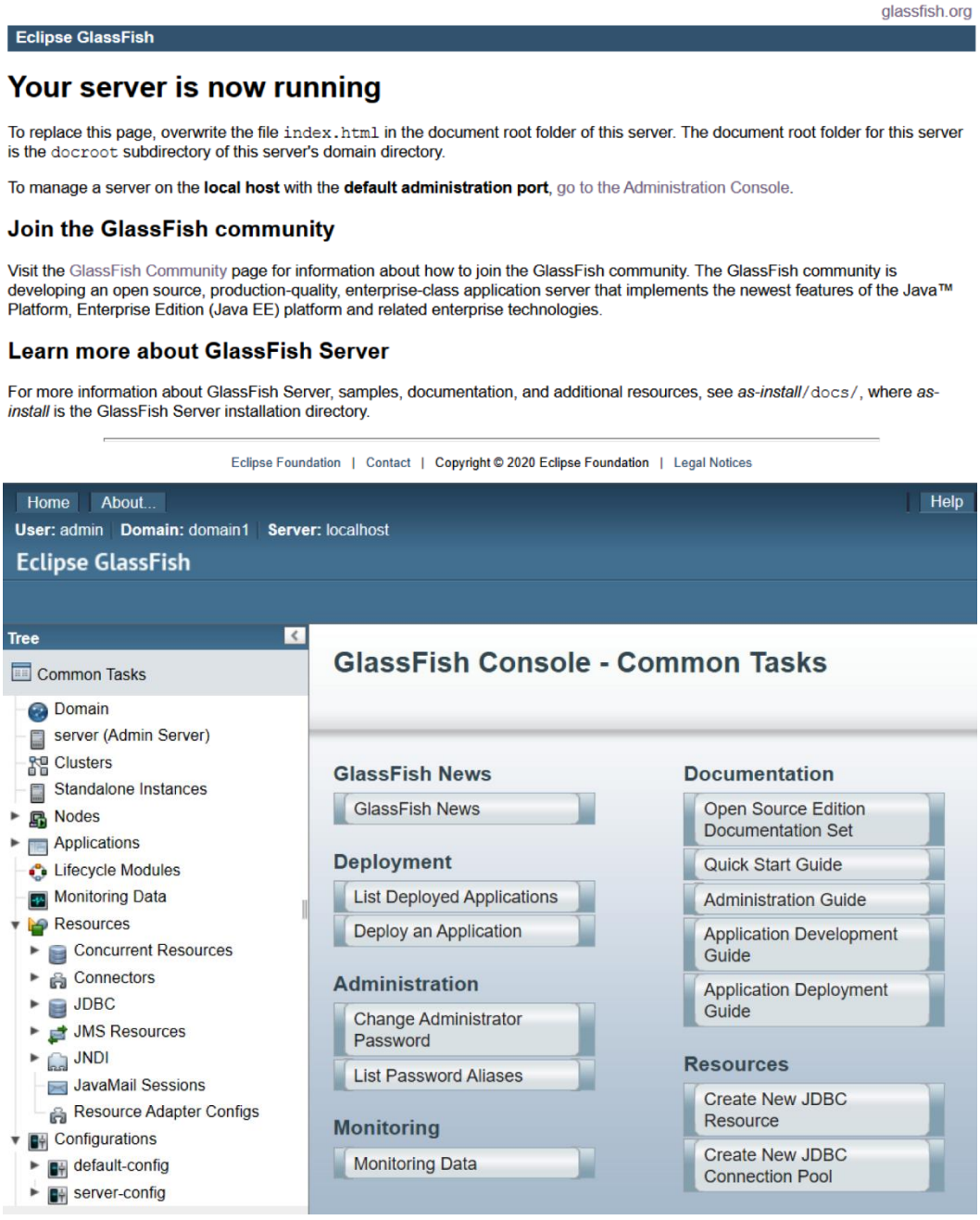

Calculs - Swing
Prêt
Epargne
Objectif
Calcul du prêt
Montant du prêt
0
Taux d'intérêt
0
Durée en mois
0
Résultat
0
Calculer

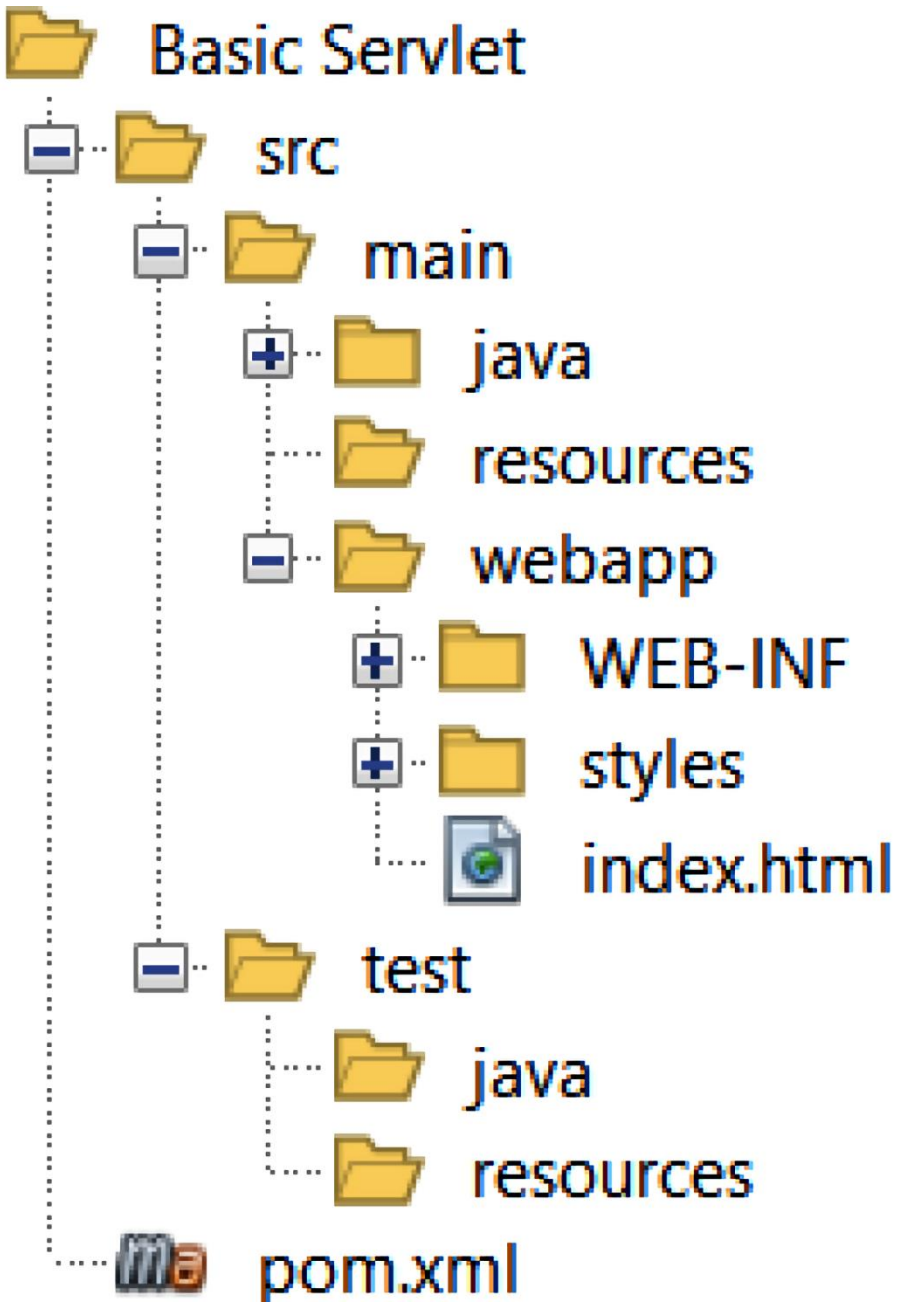

BankSwing
src
main
java
com
kenfogel
bankswing
business
FinanceCalculations.java
data
FinanceBean.java
filter
NumberDocumentFilter.java
presentation
FinanceCalculatorUI.java
FinanceCalculatorMain.java
resources
MessagesBundle.properties
MessagesBundle_en_CA.properties
MessagesBundle_fr_CA.properties
log4j2.xml
pom.xml


Calculations - Swing
Loan
Savings
Goal
Loan Calculation
Loan Amount
5000
Interest Rate
0.05
Term In Months
60
Result
94.36
Calculate


BankFX
src
main
java
com
kenfogel
bankfx
business
FinanceCalculations.java
control
BigDecimalTextField.java
converter
StringBigDecimalConverter.java
data
FinanceFXBean.java
presentation
FinanceCalculatorUI.java
FinanceCalculatorMain.java
resources
styles
Styles.css
MessagesBundle.properties
MessagesBundle_en_CA.properties
MessagesBundle_fr_CA.properties
log4j2.xml
pom.xml

Chapter 14: Server-Side Coding with Jakarta





Join our email list

To join our email list, enter your name and email address below.

Email: moose@moose.com

First Name: Ken

Last Name: Fogel

Join Now

Thanks for joining our email list

Here is the information that you entered:

Email:	moose@moose.com
First Name:	Ken
Last Name:	Fogel

To enter another email address, click on the Return button shown below.

Return

This email address was added to our list on 2023-02-05

Financial Calculator

◉ Loan ○ Savings ○ Goal

Loan Amount	5000
Interest Rate	0.05
Term In Months	60
Result	94.36

Calculate Reset

Below the min value of 1.00

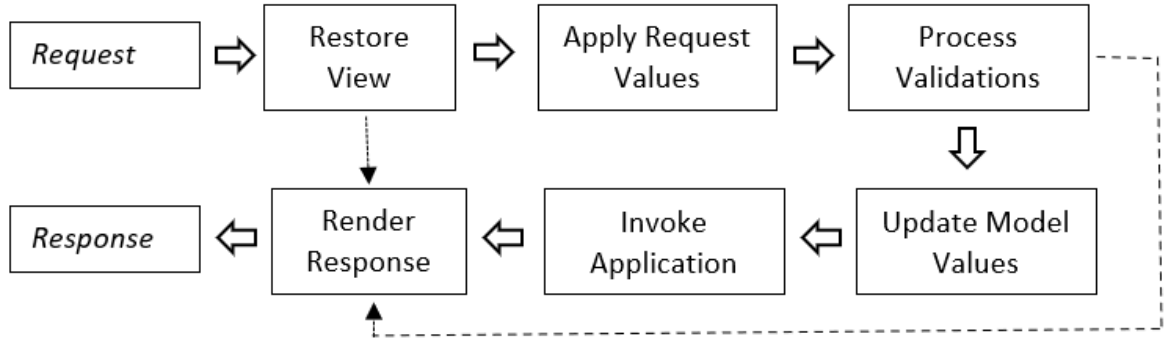
Loan Amount	-10

Your input cannot be converted to a number

Interest Rate	moose

Deployment

- List Deployed Applications
- Deploy an Application



Chapter 16: Deploying Java in Standalone Packages and Containers

No images…