CH15 練習

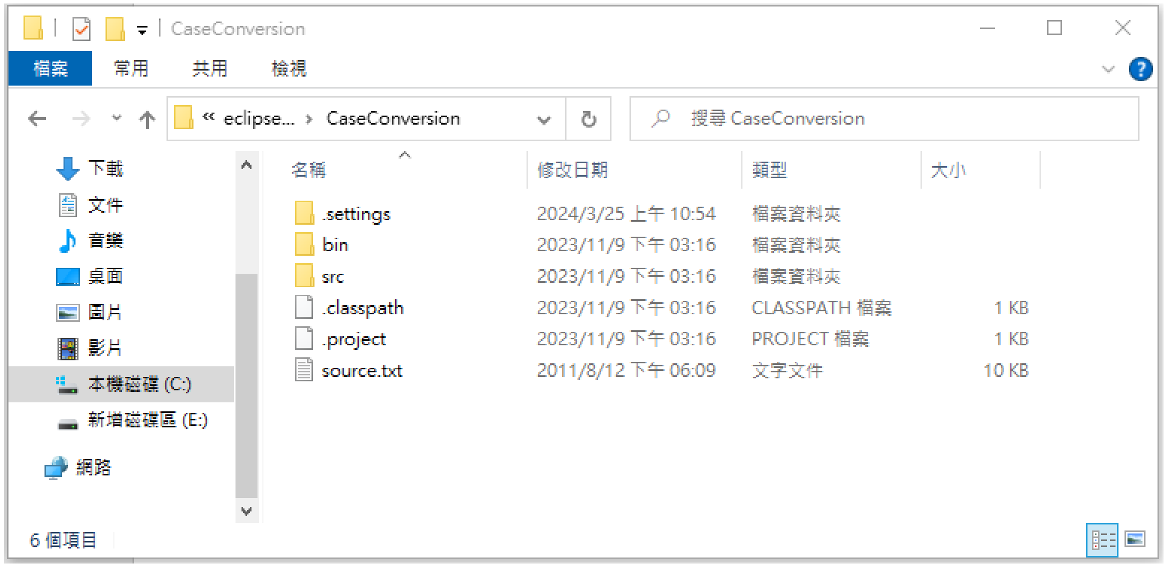
鄭安翔

ansel_cheng@hotmail.com

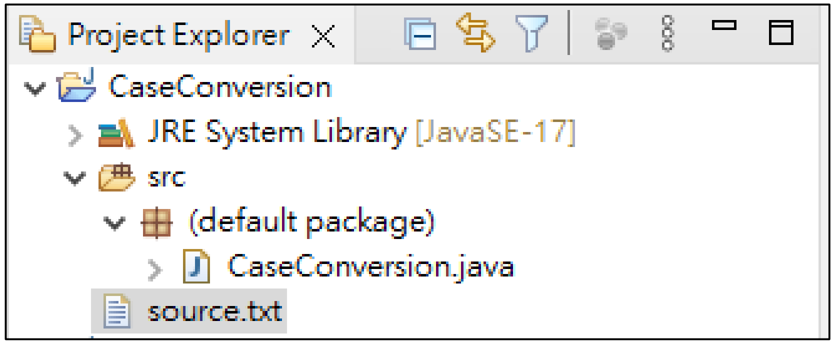
練習1 檔案大小寫轉換

- caseconversion專案
  - 複製source.txt檔案至專案路徑
  - CaseConversion程式
    - 第一個命令列參數為轉換型態
      - -U：將檔案裡的所有字母轉換成大寫
      - -L ：將檔案裡的所有字母轉換成小寫
    - 建立輸入資料流FileReader, 來源檔名為source.txt
    - 建立輸出資料流, 目的檔名為result.txt
    - 讀取來源檔案內容，轉換大小寫後，寫至目的檔案
  - 設定主類別命令列參數

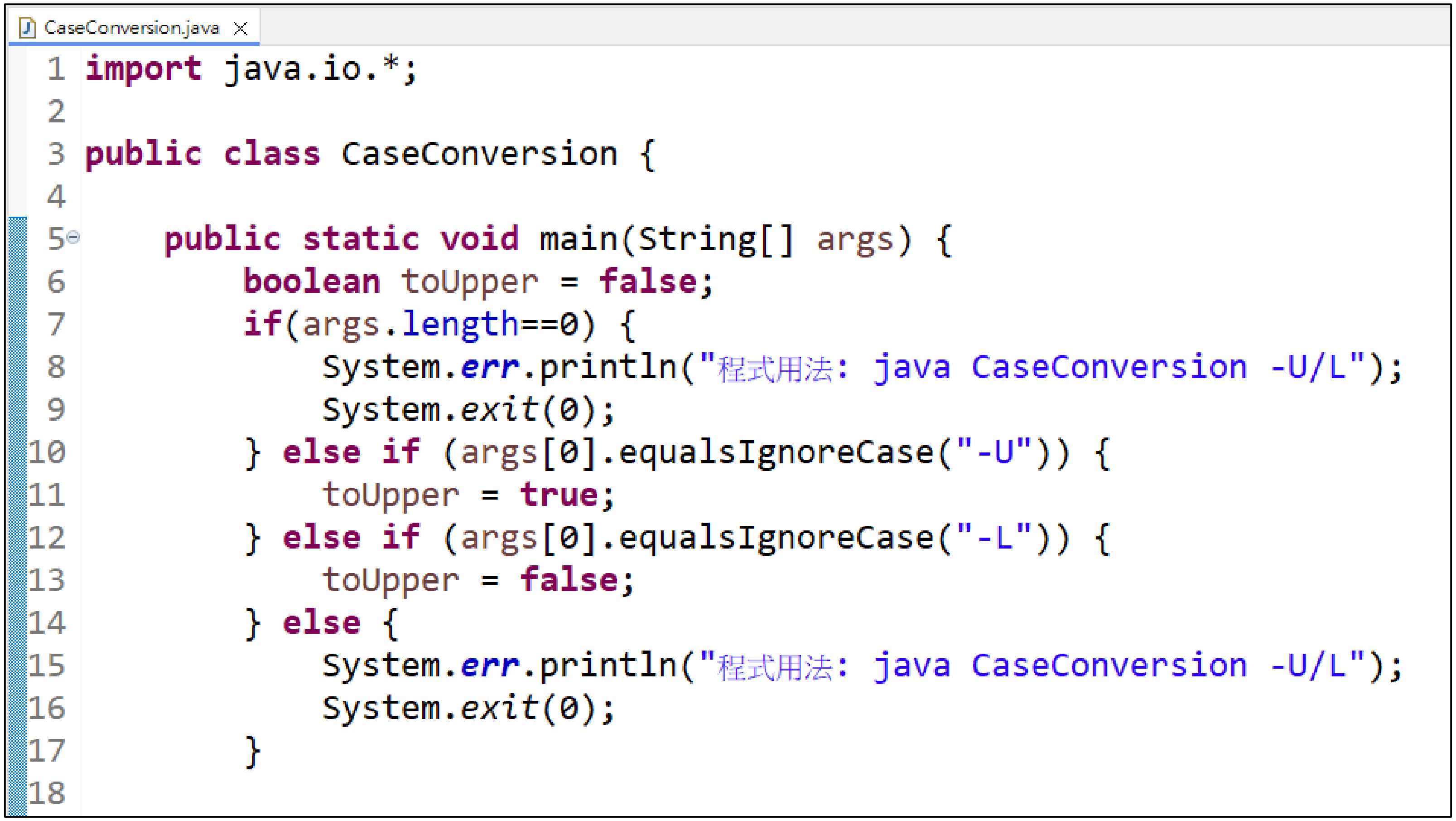
複製檔案置專案路徑



名稱	修改日期	類型	大小
.settings	2024/3/25 上午 10:54	檔案資料夾	
bin	2023/11/9 下午 03:16	檔案資料夾	
src	2023/11/9 下午 03:16	檔案資料夾	
.classpath	2023/11/9 下午 03:16	CLASSPATH 檔案	1 KB
.project	2023/11/9 下午 03:16	PROJECT 檔案	1 KB
source.txt	2011/8/12 下午 06:09	文字文件	10 KB



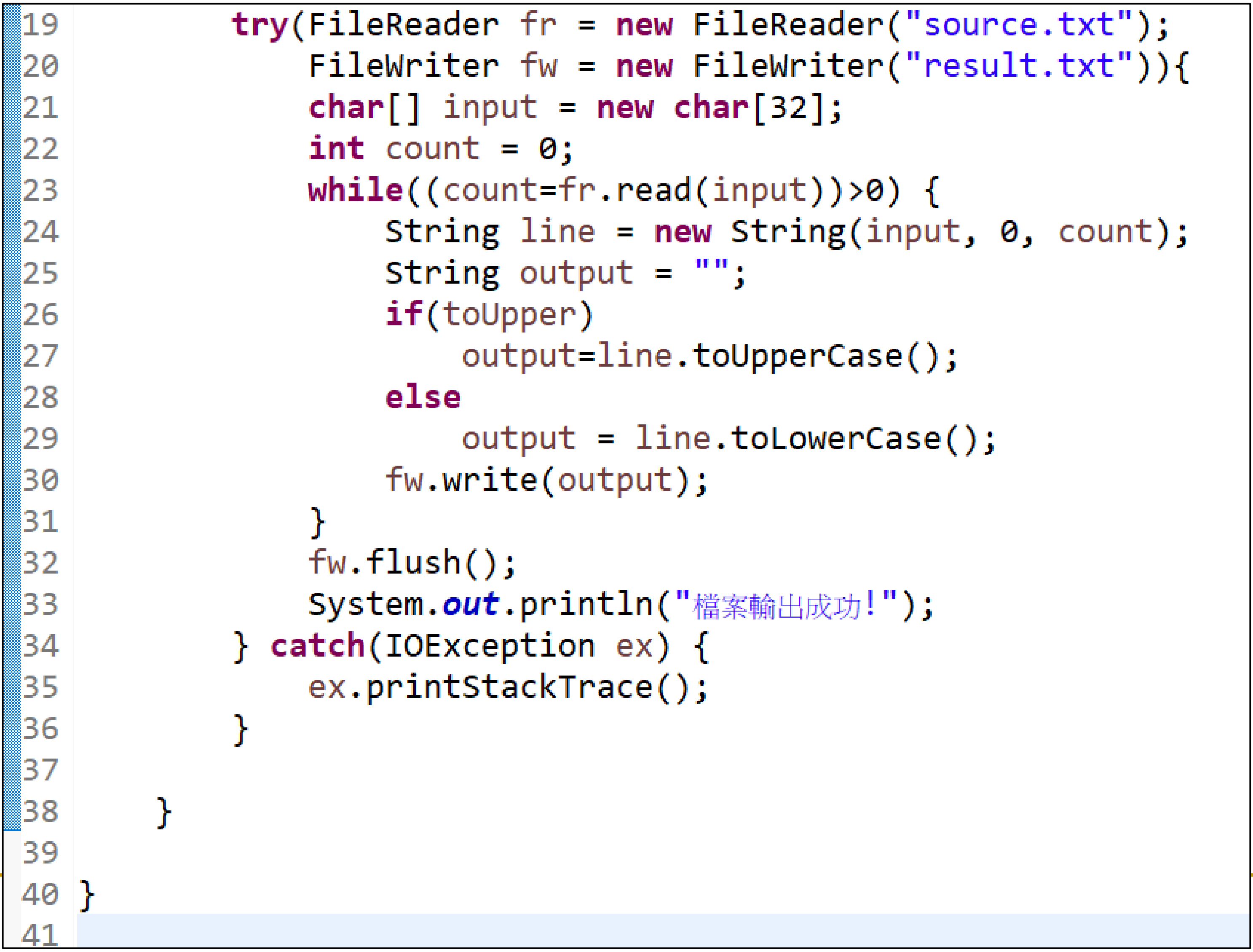
CaseConversion 類別



CaseConversion.java

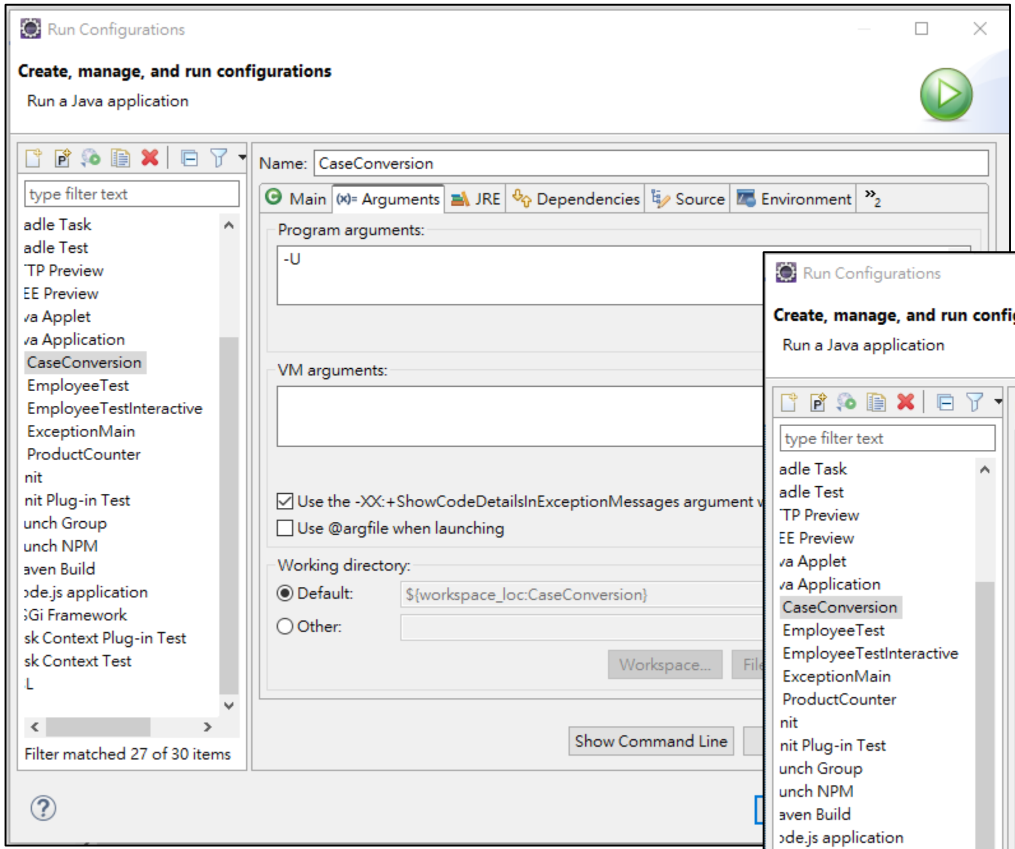
```java
import java.io.*;

public class CaseConversion {

    public static void main(String[] args) {
        boolean toUpper = false;
        if(args.length==0) {
            System.err.println("程式用法: java CaseConversion -U/L");
            System.exit(0);
        } else if (args[0].equalsIgnoreCase("-U")) {
            toUpper = true;
        } else if (args[0].equalsIgnoreCase("-L")) {
            toUpper = false;
        } else {
            System.err.println("程式用法: java CaseConversion -U/L");
            System.exit(0);
        }

```

CaseConversion 類別



```
        try(FileReader fr = new FileReader("source.txt");
            FileWriter fw = new FileWriter("result.txt")){
            char[] input = new char[32];
            int count = 0;
            while((count=fr.read(input))>0) {
                String line = new String(input, 0, count);
                String output = "";
                if(toUpper)
                    output=line.toUpperCase();
                else
                    output = line.toLowerCase();
                fw.write(output);
            }
            fw.flush();
            System.out.println("檔案輸出成功!");
        } catch(IOException ex) {
            ex.printStackTrace();
        }

    }

}
```

設定主類別傳入參數



Run Configurations

Create, manage, and run configurations

Run a Java application

type filter text

adle Task
adle Test
TP Preview
EE Preview
va Applet
va Application
CaseConversion
EmployeeTest
EmployeeTestInteractive
ExceptionMain
ProductCounter
nit
nit Plug-in Test
unch Group
unch NPM
aven Build
ode.js application
Gi Framework
sk Context Plug-in Test
sk Context Test
L

Filter matched 27 of 30 items

Name: CaseConversion

Main | Arguments | JRE | Dependencies | Source | Environment | »2

Program arguments:
```
-U
```

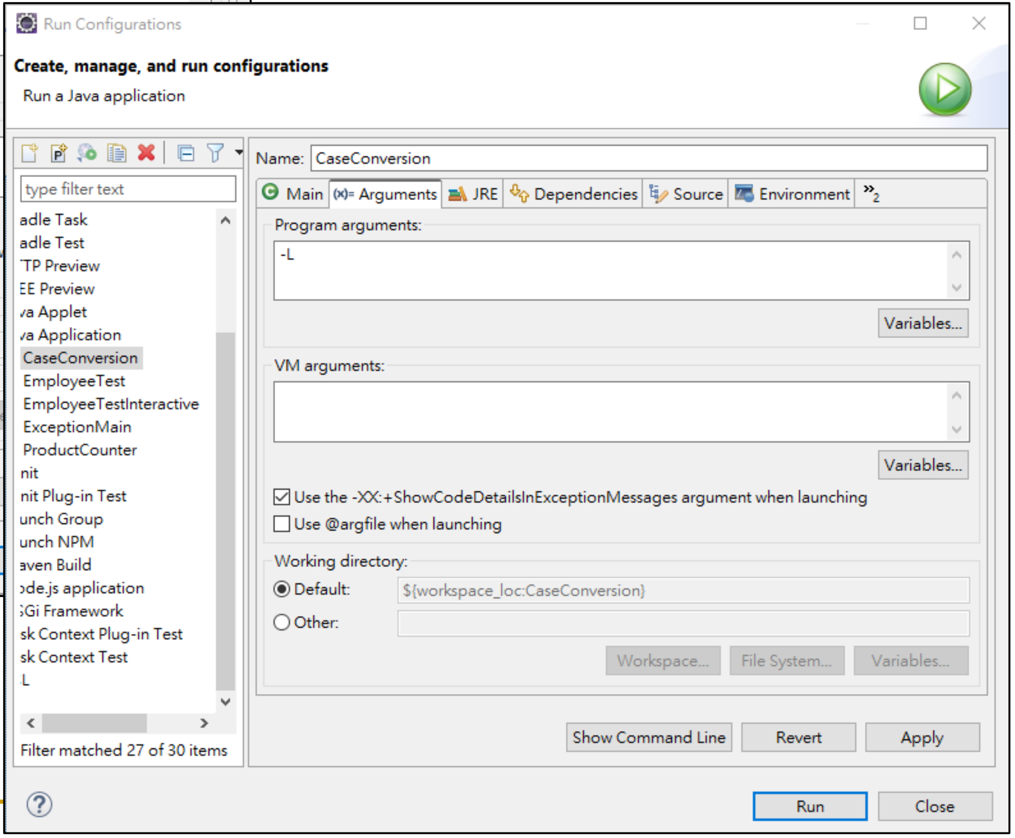
VM arguments:

[✓] Use the -XX:+ShowCodeDetailsInExceptionMessages argument w
[ ] Use @argfile when launching

Working directory:
(●) Default: ${workspace_loc:CaseConversion}
( ) Other:

Workspace... | Fil

Show Command Line



Run Configurations

Create, manage, and run configurations

Run a Java application

type filter text

adle Task
adle Test
TP Preview
EE Preview
va Applet
va Application
CaseConversion
EmployeeTest
EmployeeTestInteractive
ExceptionMain
ProductCounter
nit
nit Plug-in Test
unch Group
unch NPM
aven Build
ode.js application
Gi Framework
sk Context Plug-in Test
sk Context Test
L

Filter matched 27 of 30 items

Name: CaseConversion

Main | Arguments | JRE | Dependencies | Source | Environment | »2

Program arguments:
```
-L
```
Variables...

VM arguments:

Variables...

[✓] Use the -XX:+ShowCodeDetailsInExceptionMessages argument when launching
[ ] Use @argfile when launching

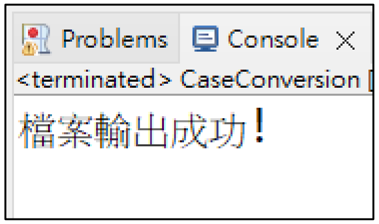
Working directory:
(●) Default: ${workspace_loc:CaseConversion}
( ) Other:

Workspace... | File System... | Variables...

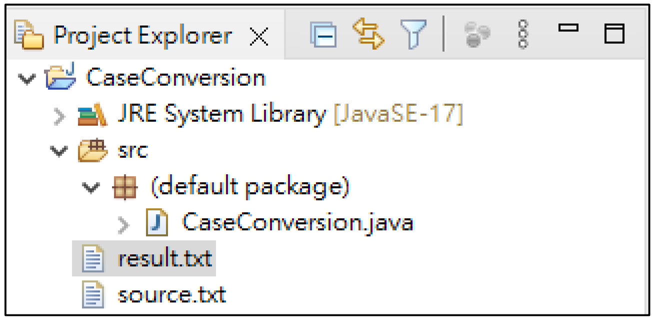
Show Command Line | Revert | Apply

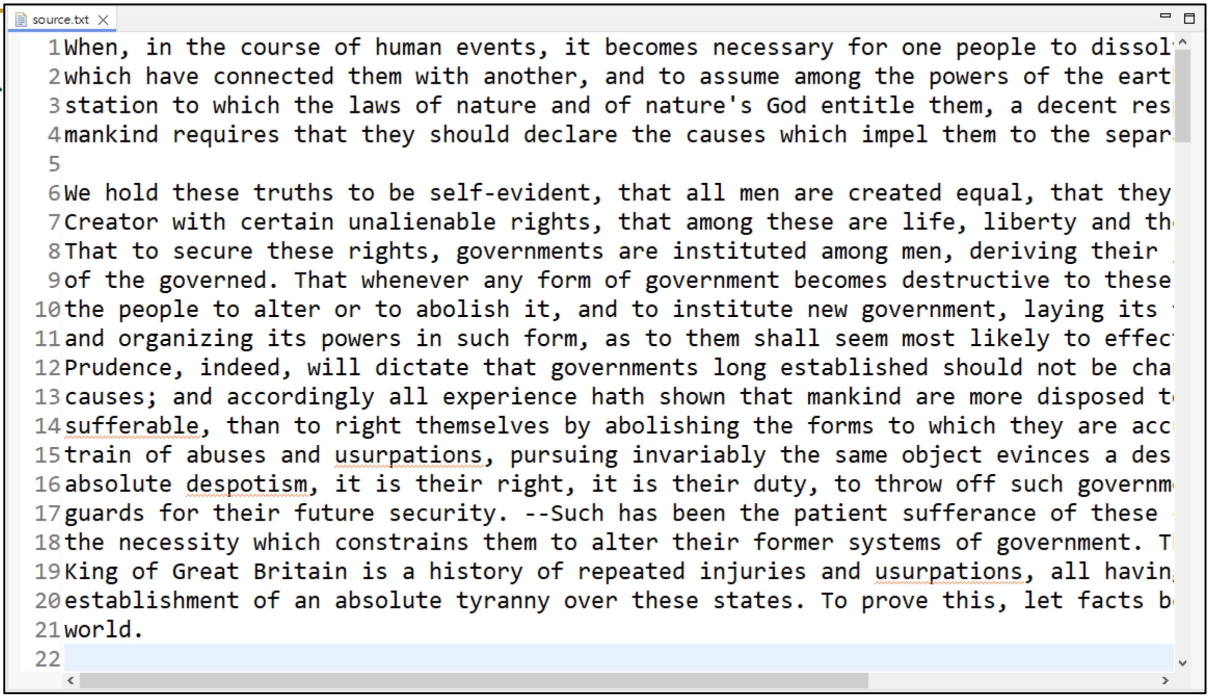
Run | Close

測試、執行

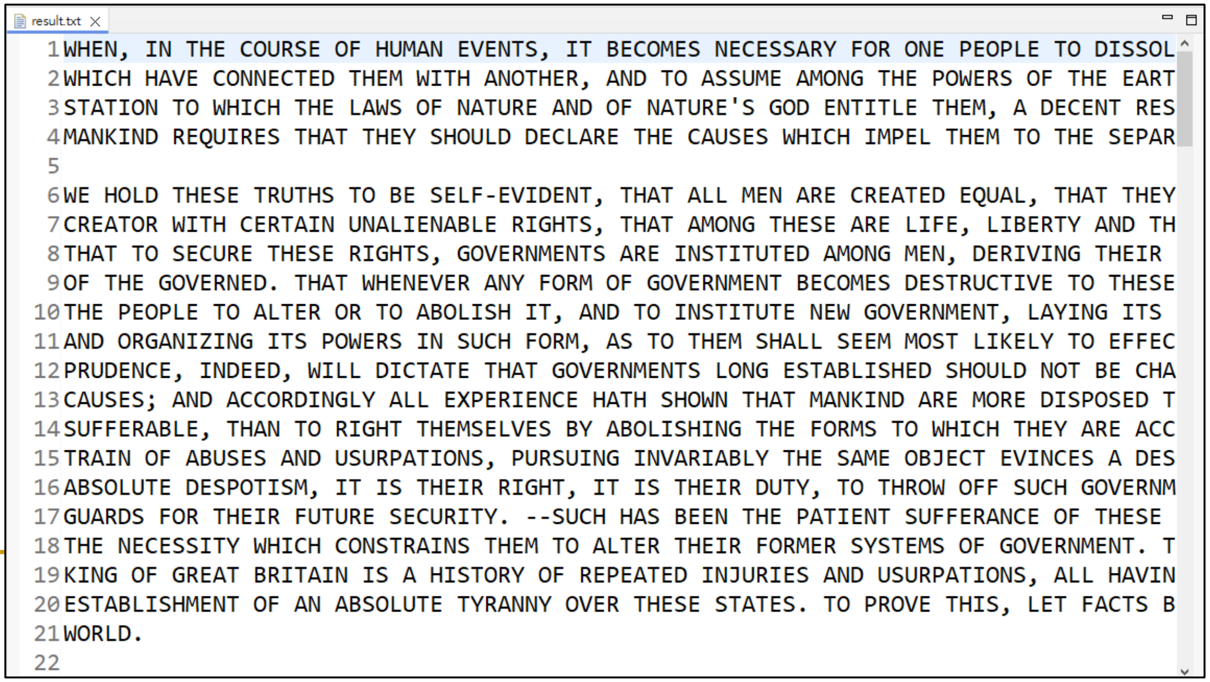


```
<terminated> CaseConversion [
檔案輸出成功!
```



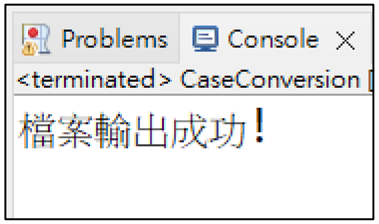


```
When, in the course of human events, it becomes necessary for one people to dissol
which have connected them with another, and to assume among the powers of the eart
station to which the laws of nature and of nature's God entitle them, a decent res
mankind requires that they should declare the causes which impel them to the separ

We hold these truths to be self-evident, that all men are created equal, that they
Creator with certain unalienable rights, that among these are life, liberty and th
That to secure these rights, governments are instituted among men, deriving their
of the governed. That whenever any form of government becomes destructive to these
the people to alter or to abolish it, and to institute new government, laying its
and organizing its powers in such form, as to them shall seem most likely to effec
Prudence, indeed, will dictate that governments long established should not be cha
causes; and accordingly all experience hath shown that mankind are more disposed t
sufferable, than to right themselves by abolishing the forms to which they are acc
train of abuses and usurpations, pursuing invariably the same object evinces a des
absolute despotism, it is their right, it is their duty, to throw off such governm
guards for their future security. --Such has been the patient sufferance of these
the necessity which constrains them to alter their former systems of government. T
King of Great Britain is a history of repeated injuries and usurpations, all havin
establishment of an absolute tyranny over these states. To prove this, let facts b
world.

```

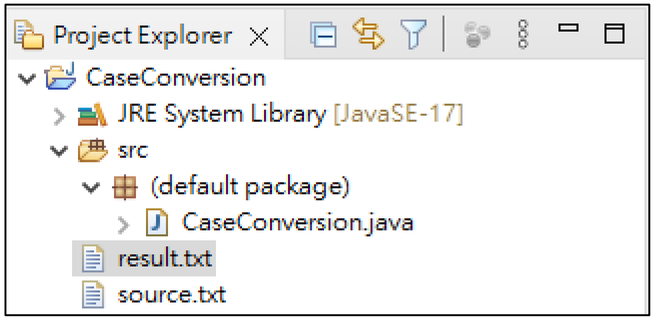


```
WHEN, IN THE COURSE OF HUMAN EVENTS, IT BECOMES NECESSARY FOR ONE PEOPLE TO DISSOL
WHICH HAVE CONNECTED THEM WITH ANOTHER, AND TO ASSUME AMONG THE POWERS OF THE EART
STATION TO WHICH THE LAWS OF NATURE AND OF NATURE'S GOD ENTITLE THEM, A DECENT RES
MANKIND REQUIRES THAT THEY SHOULD DECLARE THE CAUSES WHICH IMPEL THEM TO THE SEPAR

WE HOLD THESE TRUTHS TO BE SELF-EVIDENT, THAT ALL MEN ARE CREATED EQUAL, THAT THEY
CREATOR WITH CERTAIN UNALIENABLE RIGHTS, THAT AMONG THESE ARE LIFE, LIBERTY AND TH
THAT TO SECURE THESE RIGHTS, GOVERNMENTS ARE INSTITUTED AMONG MEN, DERIVING THEIR
OF THE GOVERNED. THAT WHENEVER ANY FORM OF GOVERNMENT BECOMES DESTRUCTIVE TO THESE
THE PEOPLE TO ALTER OR TO ABOLISH IT, AND TO INSTITUTE NEW GOVERNMENT, LAYING ITS
AND ORGANIZING ITS POWERS IN SUCH FORM, AS TO THEM SHALL SEEM MOST LIKELY TO EFFEC
PRUDENCE, INDEED, WILL DICTATE THAT GOVERNMENTS LONG ESTABLISHED SHOULD NOT BE CHA
CAUSES; AND ACCORDINGLY ALL EXPERIENCE HATH SHOWN THAT MANKIND ARE MORE DISPOSED T
SUFFERABLE, THAN TO RIGHT THEMSELVES BY ABOLISHING THE FORMS TO WHICH THEY ARE ACC
TRAIN OF ABUSES AND USURPATIONS, PURSUING INVARIABLY THE SAME OBJECT EVINCES A DES
ABSOLUTE DESPOTISM, IT IS THEIR RIGHT, IT IS THEIR DUTY, TO THROW OFF SUCH GOVERNM
GUARDS FOR THEIR FUTURE SECURITY. --SUCH HAS BEEN THE PATIENT SUFFERANCE OF THESE
THE NECESSITY WHICH CONSTRAINS THEM TO ALTER THEIR FORMER SYSTEMS OF GOVERNMENT. T
KING OF GREAT BRITAIN IS A HISTORY OF REPEATED INJURIES AND USURPATIONS, ALL HAVIN
ESTABLISHMENT OF AN ABSOLUTE TYRANNY OVER THESE STATES. TO PROVE THIS, LET FACTS B
WORLD.

```

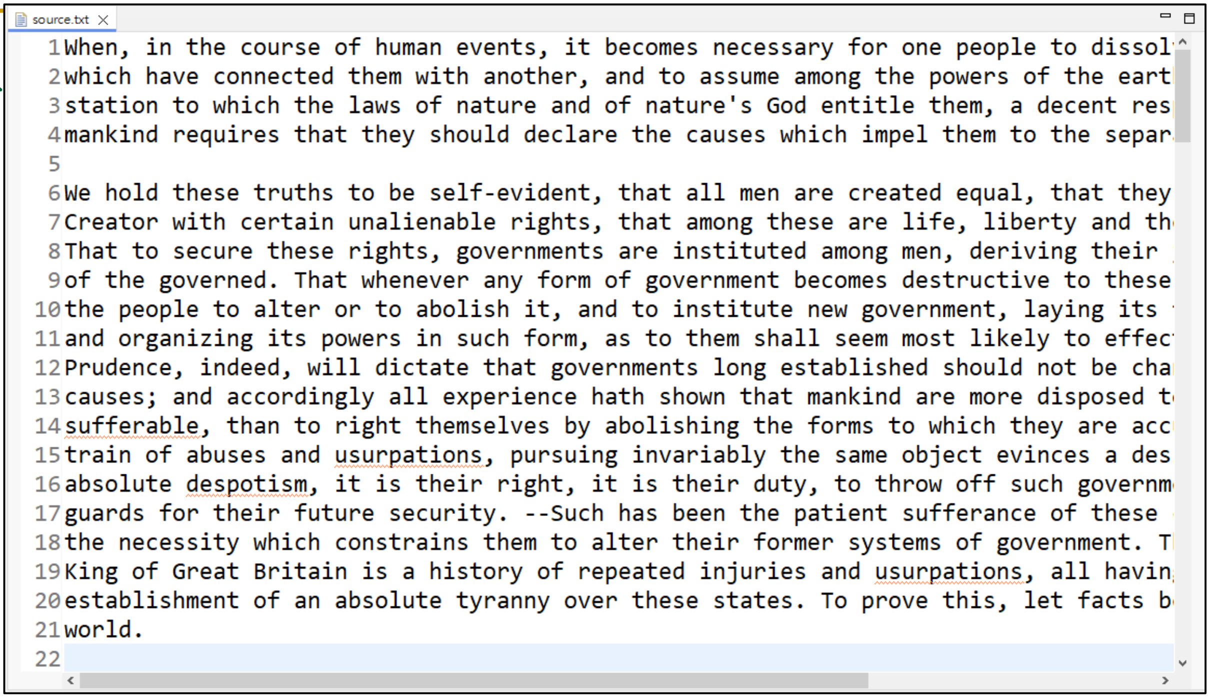
測試、執行



```
<terminated> CaseConversion
檔案輸出成功!
```

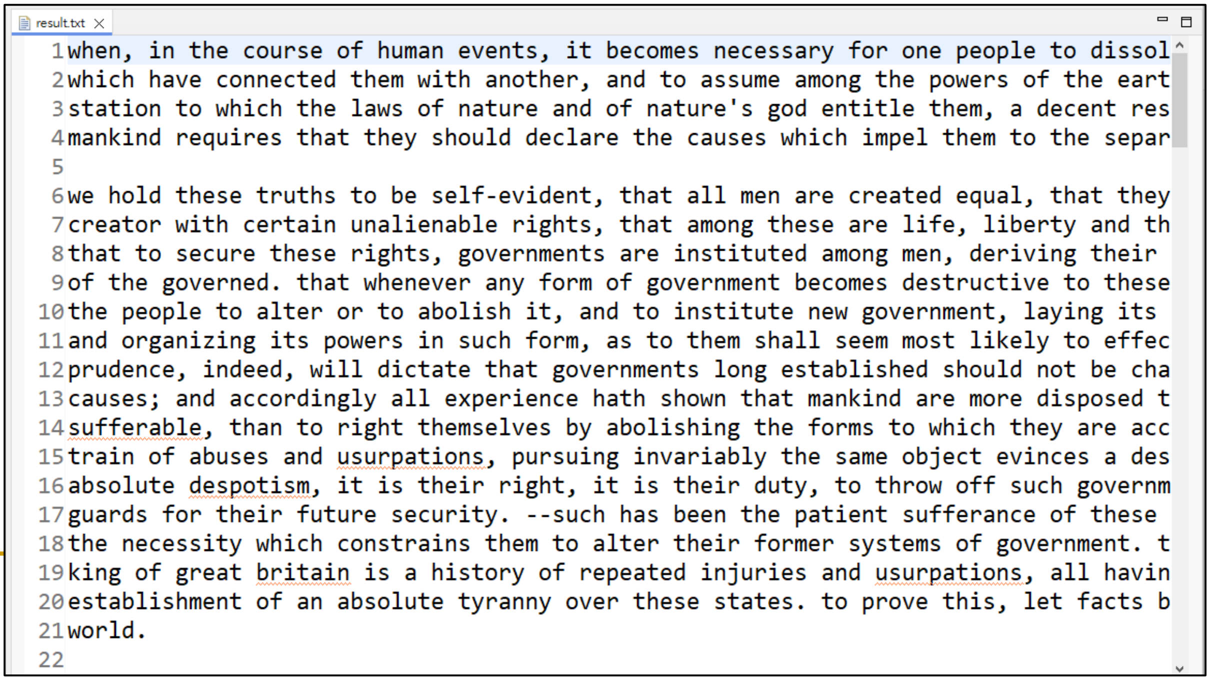


```
CaseConversion
  JRE System Library [JavaSE-17]
  src
    (default package)
      CaseConversion.java
  result.txt
  source.txt
```



source.txt

```
When, in the course of human events, it becomes necessary for one people to dissol
which have connected them with another, and to assume among the powers of the eart
station to which the laws of nature and of nature's God entitle them, a decent res
mankind requires that they should declare the causes which impel them to the separ

We hold these truths to be self-evident, that all men are created equal, that they
Creator with certain unalienable rights, that among these are life, liberty and th
That to secure these rights, governments are instituted among men, deriving their
of the governed. That whenever any form of government becomes destructive to these
the people to alter or to abolish it, and to institute new government, laying its
and organizing its powers in such form, as to them shall seem most likely to effec
Prudence, indeed, will dictate that governments long established should not be cha
causes; and accordingly all experience hath shown that mankind are more disposed t
sufferable, than to right themselves by abolishing the forms to which they are acc
train of abuses and usurpations, pursuing invariably the same object evinces a des
absolute despotism, it is their right, it is their duty, to throw off such governm
guards for their future security. --Such has been the patient sufferance of these
the necessity which constrains them to alter their former systems of government. T
King of Great Britain is a history of repeated injuries and usurpations, all havin
establishment of an absolute tyranny over these states. To prove this, let facts b
world.

```



result.txt

```
when, in the course of human events, it becomes necessary for one people to dissol
which have connected them with another, and to assume among the powers of the eart
station to which the laws of nature and of nature's god entitle them, a decent res
mankind requires that they should declare the causes which impel them to the separ

we hold these truths to be self-evident, that all men are created equal, that they
creator with certain unalienable rights, that among these are life, liberty and th
that to secure these rights, governments are instituted among men, deriving their
of the governed. that whenever any form of government becomes destructive to these
the people to alter or to abolish it, and to institute new government, laying its
and organizing its powers in such form, as to them shall seem most likely to effec
prudence, indeed, will dictate that governments long established should not be cha
causes; and accordingly all experience hath shown that mankind are more disposed t
sufferable, than to right themselves by abolishing the forms to which they are acc
train of abuses and usurpations, pursuing invariably the same object evinces a des
absolute despotism, it is their right, it is their duty, to throw off such governm
guards for their future security. --such has been the patient sufferance of these
the necessity which constrains them to alter their former systems of government. t
king of great britain is a history of repeated injuries and usurpations, all havin
establishment of an absolute tyranny over these states. to prove this, let facts b
world.

```

練習2 檔案實作EmployeeDAO

1. 修改 EmployeeDAO 專案
2. 新增com.example.dao.EmployeeDAOFileImpl.java
   - 實作com.example.dao.EmployeeDAO介面
   - 三個私有屬性, 員工集合以SortedMap實作
     - static SortedMap<Integer, Employee> employees = new TreeMap<>();
     - SimpleDateFormat df=new SimpleDateFormat("MMM d, yyyy", Locale.US);
     - String fileName;
   - 新增建構子
     - EmployeeDAOFileImpl(String fileName)
   - 使用employees.txt檔案紀錄員工資料
     - 資料格式 : 1|Sean|Cheng|Mar 21, 1974|50000.00

練習2 檔案實作EmployeeDAO

- 新增資料讀入同步方法syncData()
  - 以fileName屬性建立BufferedReader br, 置於try with Resource中
  - 讀取檔案資料, 一次讀一行
  - 切割資料字串, 並做適當轉型
    - split("\\|")
  - 建立對應的物件
    - Employee(int id, String firstName, String lastName, Date birthDate, float salary)
  - 以 id為key, Employee物件為value, 加入SortedMap employees中
- 新增資料寫出同步方法commit()
  - 以fileName屬性建立PrintWriter pw, 置於try with Resource中
  - 將employees中的Employee物件一一取出,
  - 依上述資料格式建立對應儲存字串
  - 將儲存字串寫入紀錄檔

練習2 檔案實作EmployeeDAO

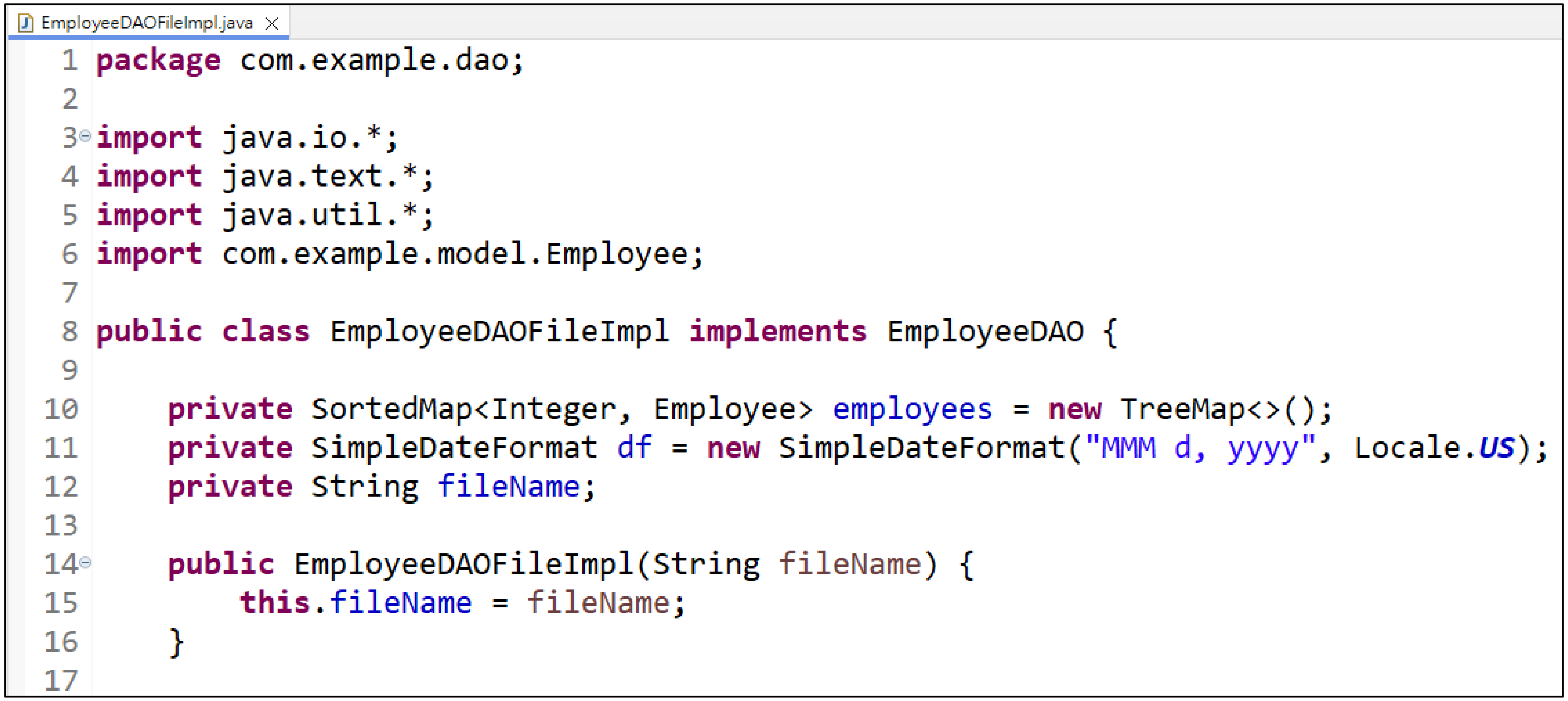
- 實作 add(emp : Employee) 方法
  - 查詢id是否不存在, 若存在傳回DAOException
  - 以 id為key, emp為value, 加入SortedMap employees中
  - 資料寫出commit()
- 實作 update(emp : Employee) 方法
  - 查詢id是否已存在, 若不存在傳回DAOException
  - 以 id為key, emp為value, 加入SortedMap employees中
  - 資料寫出commit()
- 實作 delete(id : int) 方法
  - 查詢id是否已存在, 若不存在傳回DAOException
  - SortedMap employees中移除key為id的鍵值對
  - 資料寫出commit()

練習2 檔案實作EmployeeDAO

- 實作 findById(id : int) : Employee 方法
  - 資料讀入syncData(), 取得最新的employees資料
  - 傳回鍵值為id所對應的Employee物件
- 實作 getAllEmployees(): Employee[ ] 方法
  - 資料讀入syncData(), 取得最新的employees資料
  - 將employees資料轉為Employee[ ]後傳回
- 實作 close()方法

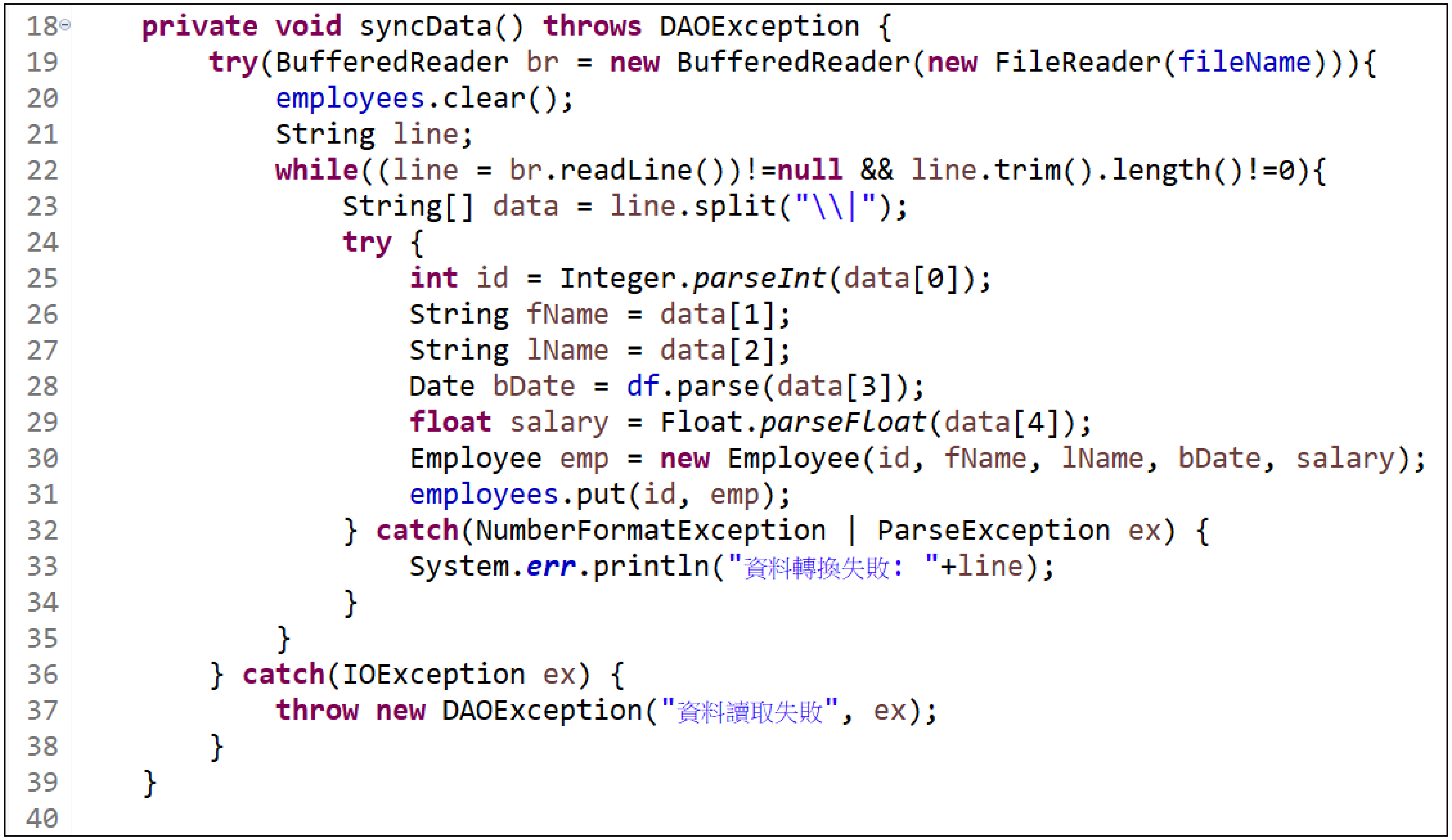
3. 修改EmployeeDAOFactory
   - 修改 createEmployeeDAO() : EmployeeDAO 方法
     - 傳回以employee.txt為傳入參數所建立之EmployeeDAOFileImpl物件
4. 刪除EmployeeDAOMemoryImpl.java
5. 測試、執行

EmployeeDAOFileImpl類別


EmployeeDAOFileImpl.java

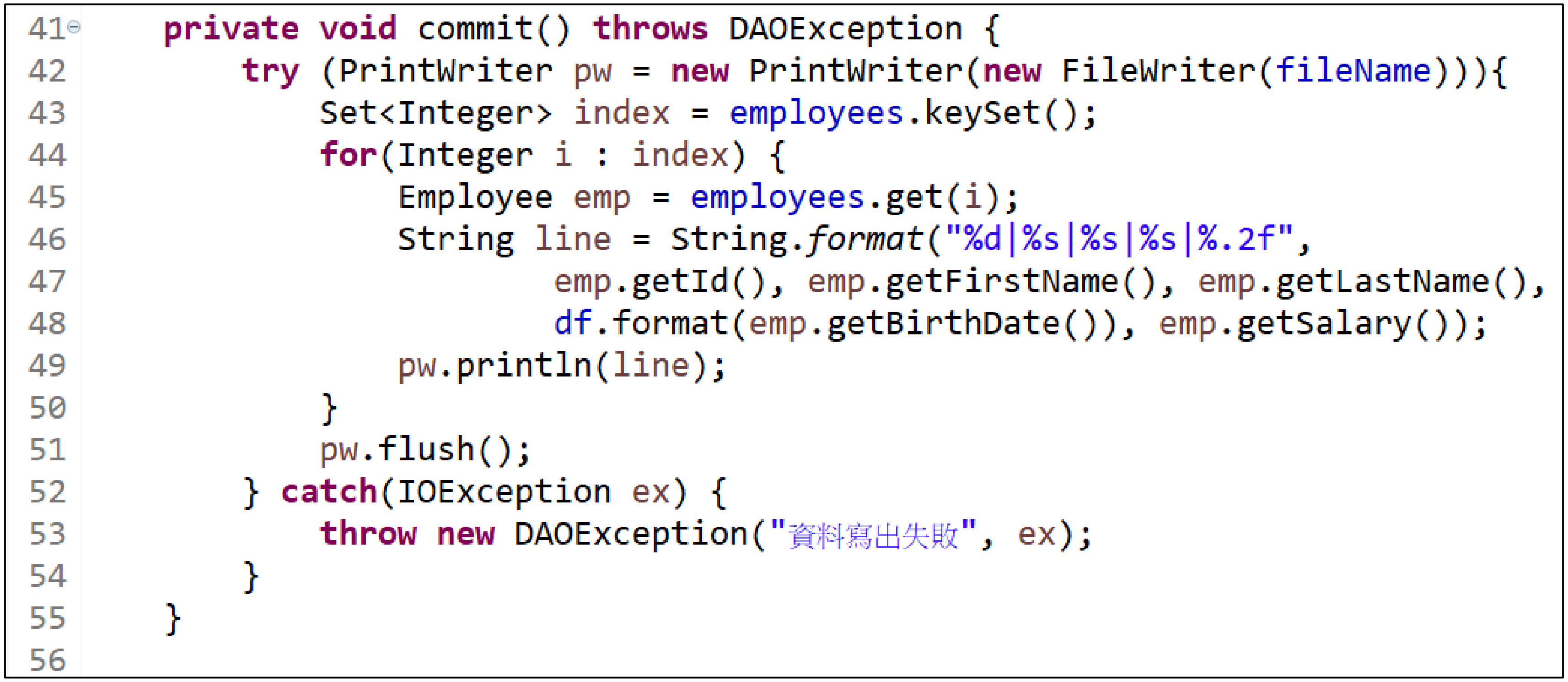
```java
package com.example.dao;

import java.io.*;
import java.text.*;
import java.util.*;
import com.example.model.Employee;

public class EmployeeDAOFileImpl implements EmployeeDAO {

    private SortedMap<Integer, Employee> employees = new TreeMap<>();
    private SimpleDateFormat df = new SimpleDateFormat("MMM d, yyyy", Locale.US);
    private String fileName;

    public EmployeeDAOFileImpl(String fileName) {
        this.fileName = fileName;
    }

```

EmployeeDAOFileImpl類別



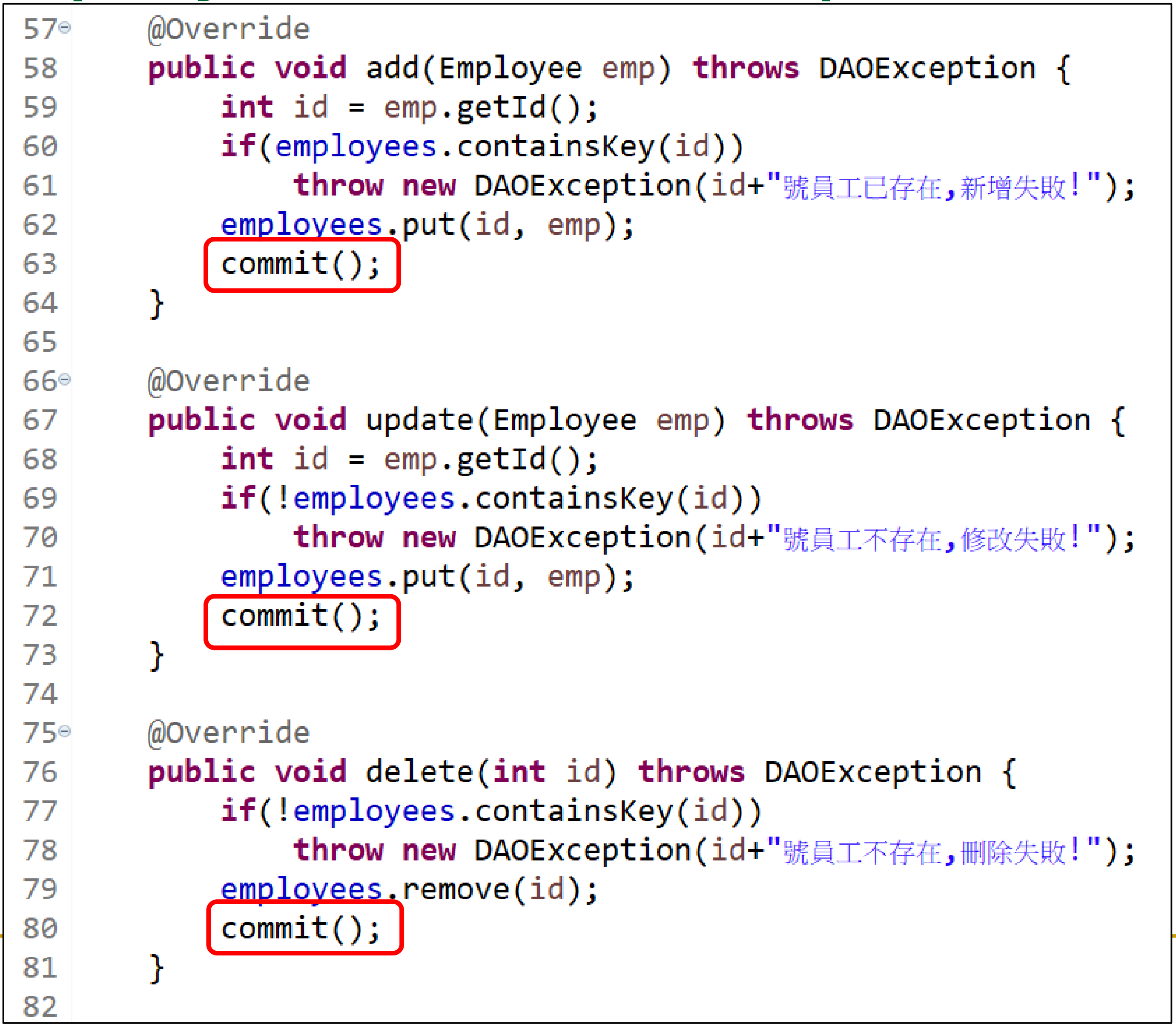
```java
    private void syncData() throws DAOException {
        try(BufferedReader br = new BufferedReader(new FileReader(fileName))){
            employees.clear();
            String line;
            while((line = br.readLine())!=null && line.trim().length()!=0){
                String[] data = line.split("\\|");
                try {
                    int id = Integer.parseInt(data[0]);
                    String fName = data[1];
                    String lName = data[2];
                    Date bDate = df.parse(data[3]);
                    float salary = Float.parseFloat(data[4]);
                    Employee emp = new Employee(id, fName, lName, bDate, salary);
                    employees.put(id, emp);
                } catch(NumberFormatException | ParseException ex) {
                    System.err.println("資料轉換失敗: "+line);
                }
            }
        } catch(IOException ex) {
            throw new DAOException("資料讀取失敗", ex);
        }
    }
```

EmployeeDAOFileImpl類別



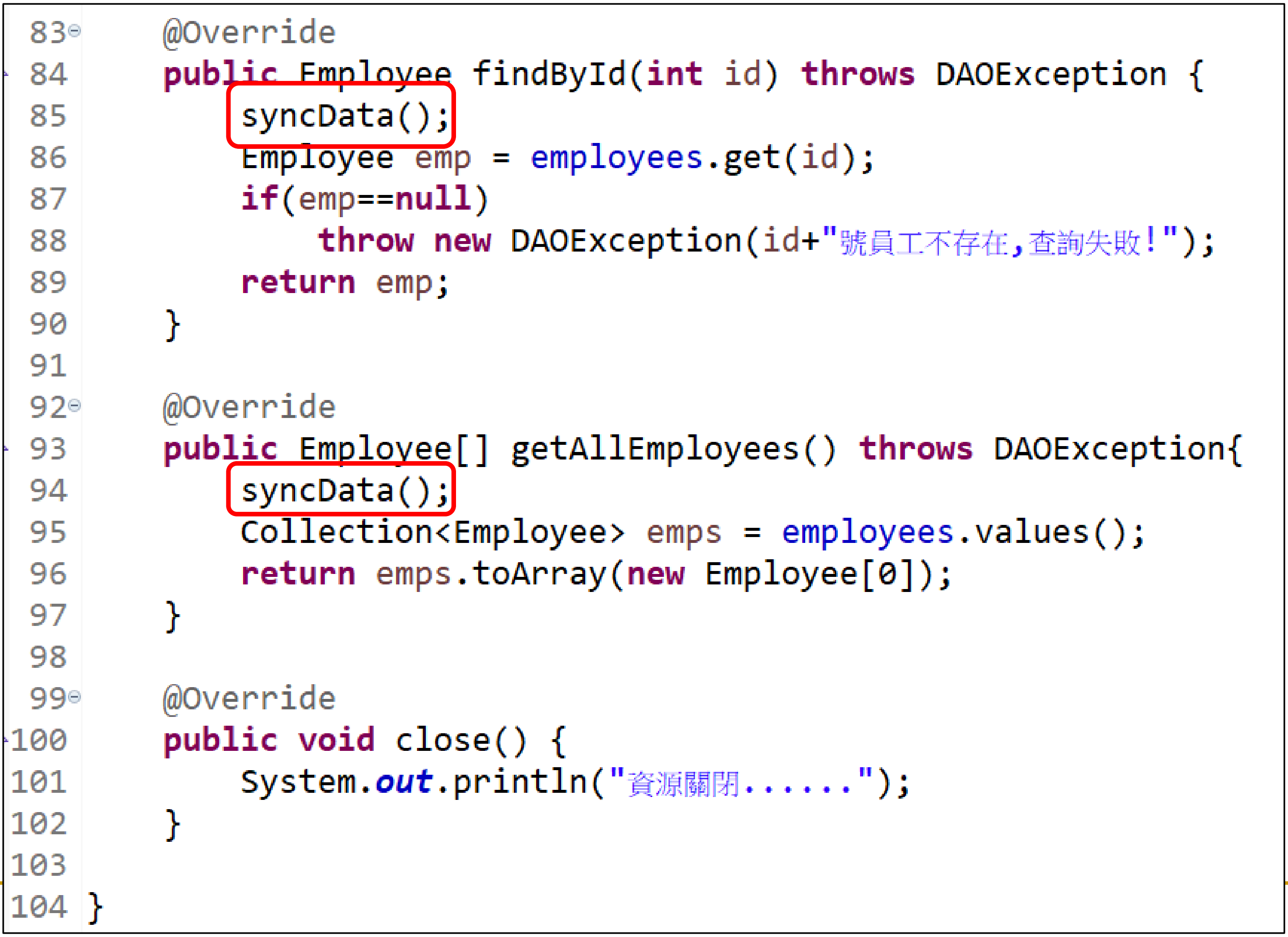
```java
    private void commit() throws DAOException {
        try (PrintWriter pw = new PrintWriter(new FileWriter(fileName))){
            Set<Integer> index = employees.keySet();
            for(Integer i : index) {
                Employee emp = employees.get(i);
                String line = String.format("%d|%s|%s|%s|%.2f",
                        emp.getId(), emp.getFirstName(), emp.getLastName(),
                        df.format(emp.getBirthDate()), emp.getSalary());
                pw.println(line);
            }
            pw.flush();
        } catch(IOException ex) {
            throw new DAOException("資料寫出失敗", ex);
        }
    }

```

EmployeeDAOFileImpl類別



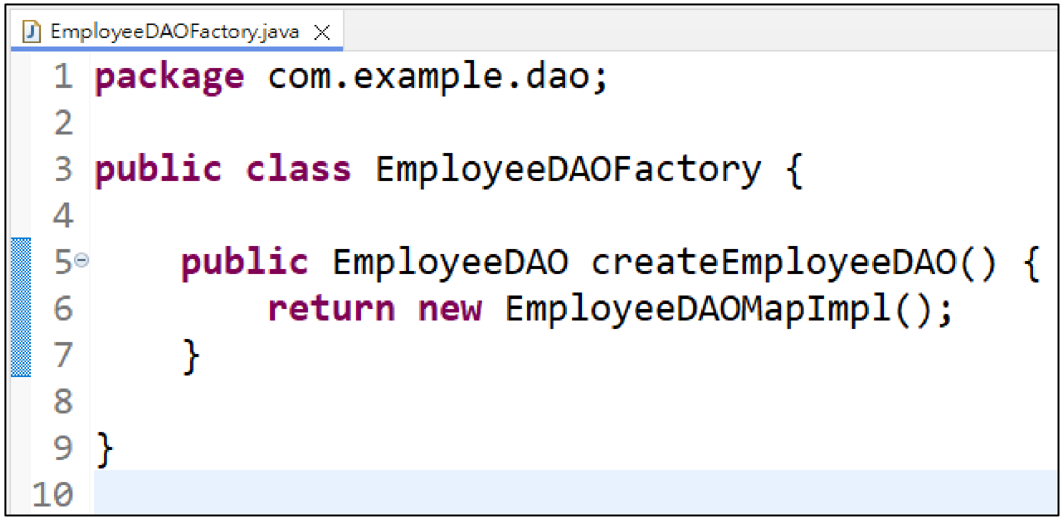
```
    @Override
    public void add(Employee emp) throws DAOException {
        int id = emp.getId();
        if(employees.containsKey(id))
            throw new DAOException(id+"號員工已存在,新增失敗!");
        employees.put(id, emp);
        commit();
    }

    @Override
    public void update(Employee emp) throws DAOException {
        int id = emp.getId();
        if(!employees.containsKey(id))
            throw new DAOException(id+"號員工不存在,修改失敗!");
        employees.put(id, emp);
        commit();
    }

    @Override
    public void delete(int id) throws DAOException {
        if(!employees.containsKey(id))
            throw new DAOException(id+"號員工不存在,刪除失敗!");
        employees.remove(id);
        commit();
    }

```

EmployeeDAOFileImpl類別



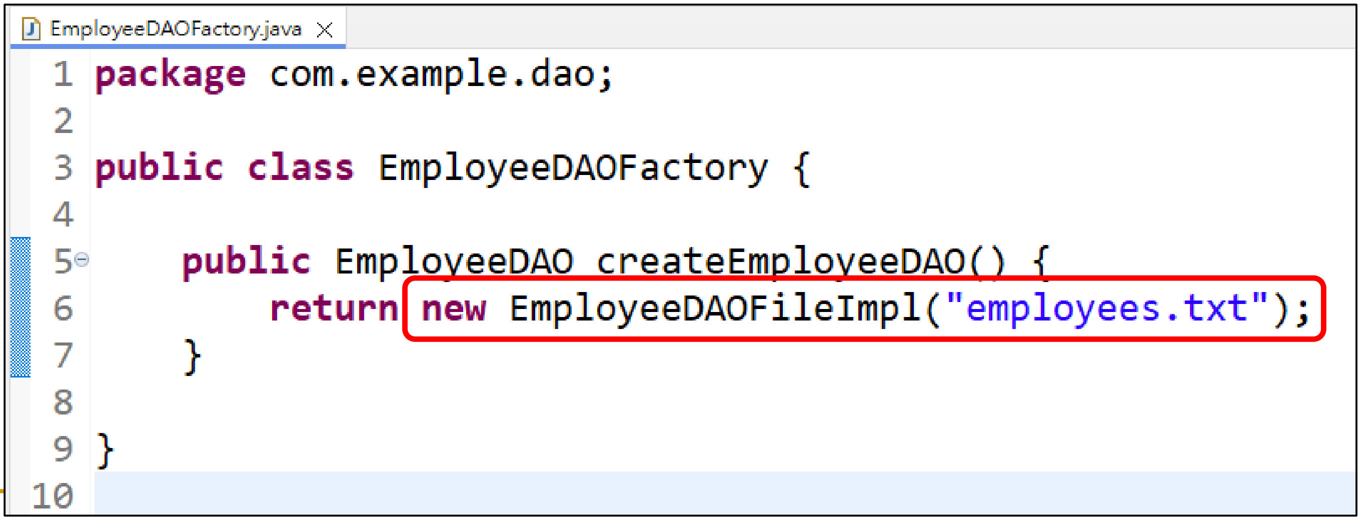
```
    @Override
    public Employee findById(int id) throws DAOException {
        syncData();
        Employee emp = employees.get(id);
        if(emp==null)
            throw new DAOException(id+"號員工不存在,查詢失敗!");
        return emp;
    }

    @Override
    public Employee[] getAllEmployees() throws DAOException{
        syncData();
        Collection<Employee> emps = employees.values();
        return emps.toArray(new Employee[0]);
    }

    @Override
    public void close() {
        System.out.println("資源關閉......");
    }

}
```

EmployeeDAOFactory 類別



EmployeeDAOFactory.java

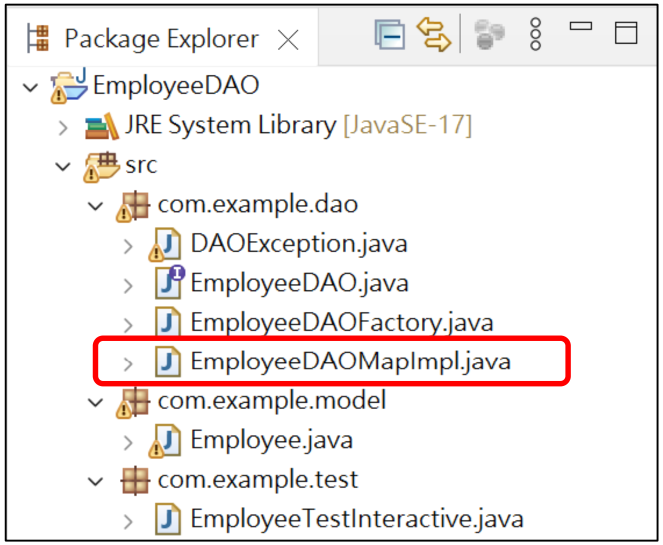
```java
package com.example.dao;

public class EmployeeDAOFactory {

    public EmployeeDAO createEmployeeDAO() {
        return new EmployeeDAOMapImpl();
    }

}
```



EmployeeDAOFactory.java

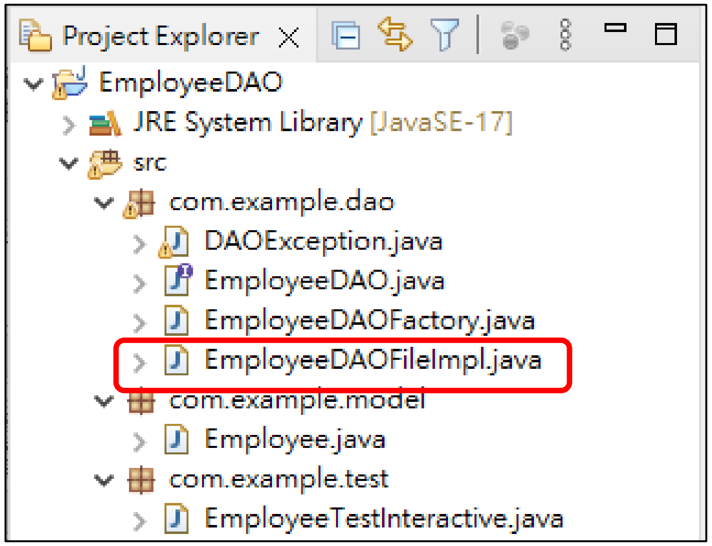
```java
package com.example.dao;

public class EmployeeDAOFactory {

    public EmployeeDAO createEmployeeDAO() {
        return new EmployeeDAOFileImpl("employees.txt");
    }

}
```

刪除EmployeeDAOMemoryImpl.java



Package Explorer

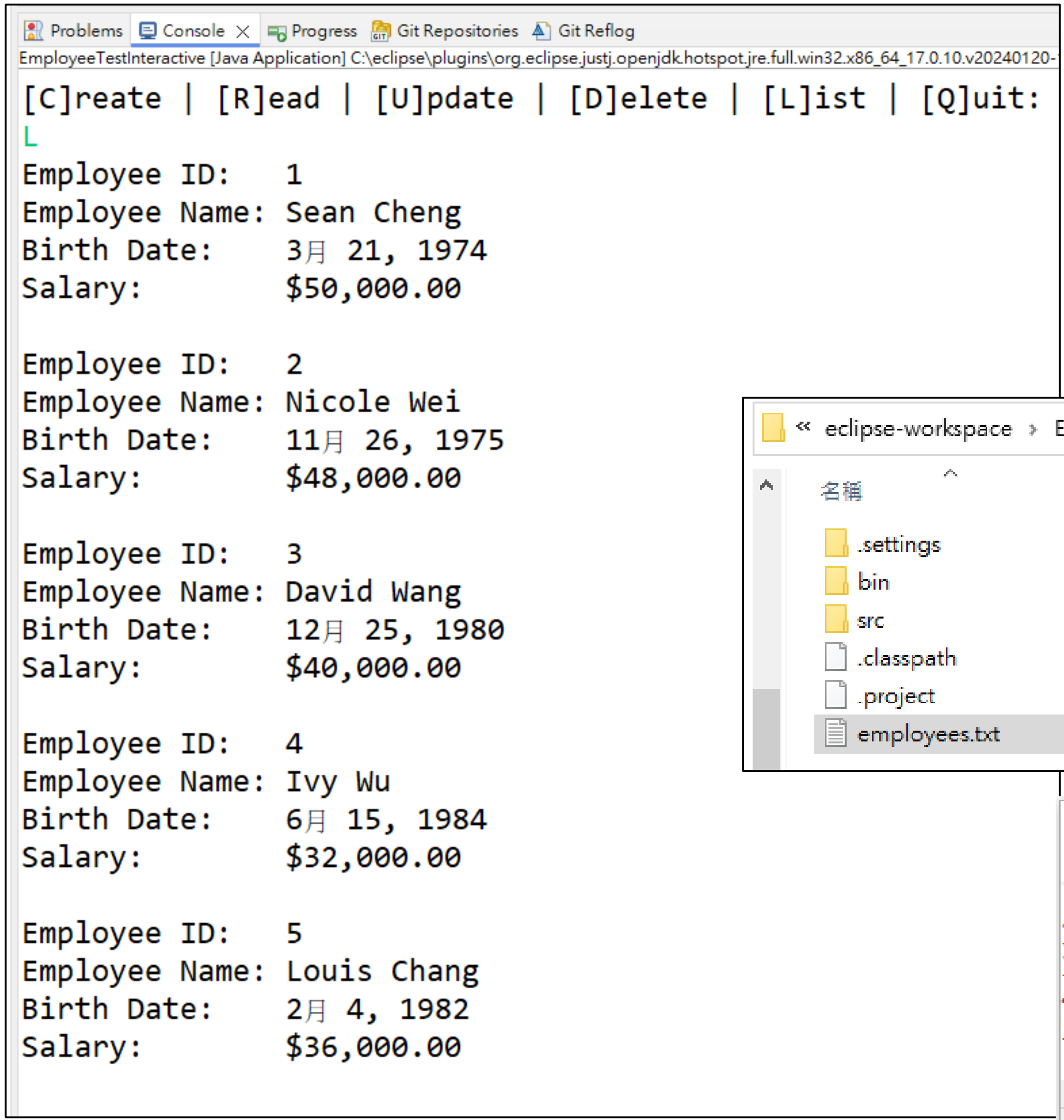
- EmployeeDAO
  - JRE System Library [JavaSE-17]
  - src
    - com.example.dao
      - DAOException.java
      - EmployeeDAO.java
      - EmployeeDAOFactory.java
      - EmployeeDAOMapImpl.java
    - com.example.model
      - Employee.java
    - com.example.test
      - EmployeeTestInteractive.java



Project Explorer

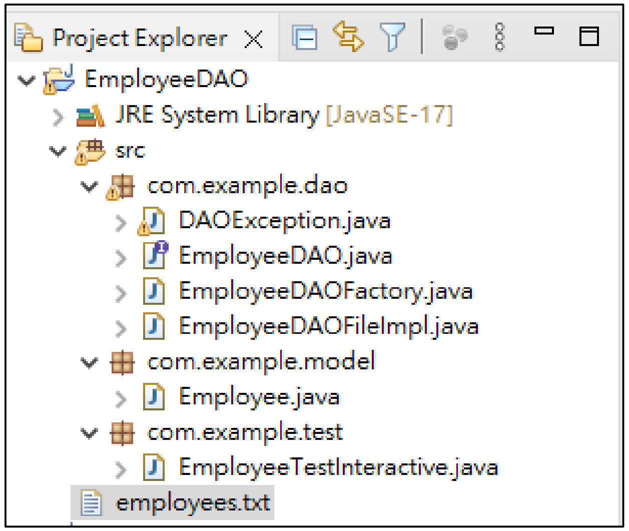
- EmployeeDAO
  - JRE System Library [JavaSE-17]
  - src
    - com.example.dao
      - DAOException.java
      - EmployeeDAO.java
      - EmployeeDAOFactory.java
      - EmployeeDAOFileImpl.java
    - com.example.model
      - Employee.java
    - com.example.test
      - EmployeeTestInteractive.java

測試、執行

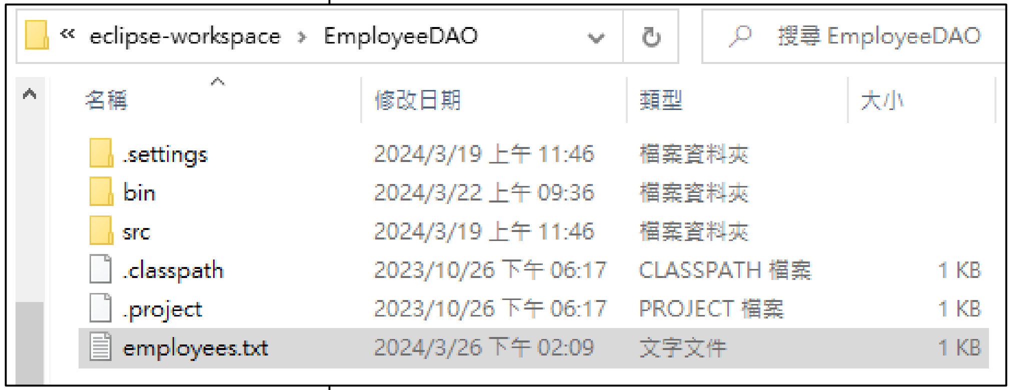


EmployeeTestInteractive [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v20240120-

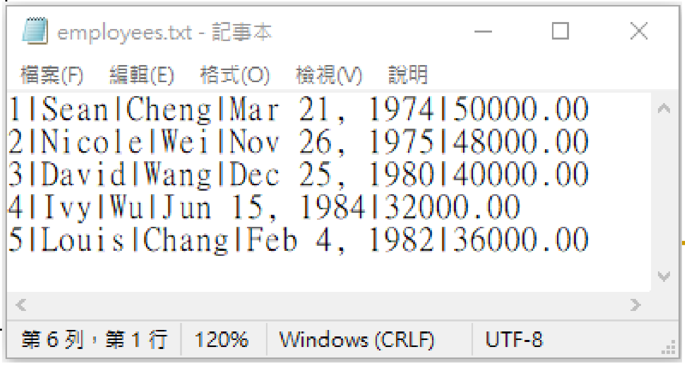
```
[C]reate | [R]ead | [U]pdate | [D]elete | [L]ist | [Q]uit:
L
Employee ID:   1
Employee Name: Sean Cheng
Birth Date:    3月 21, 1974
Salary:        $50,000.00

Employee ID:   2
Employee Name: Nicole Wei
Birth Date:    11月 26, 1975
Salary:        $48,000.00

Employee ID:   3
Employee Name: David Wang
Birth Date:    12月 25, 1980
Salary:        $40,000.00

Employee ID:   4
Employee Name: Ivy Wu
Birth Date:    6月 15, 1984
Salary:        $32,000.00

Employee ID:   5
Employee Name: Louis Chang
Birth Date:    2月 4, 1982
Salary:        $36,000.00
```



```
EmployeeDAO
  JRE System Library [JavaSE-17]
  src
    com.example.dao
      DAOException.java
      EmployeeDAO.java
      EmployeeDAOFactory.java
      EmployeeDAOFileImpl.java
    com.example.model
      Employee.java
    com.example.test
      EmployeeTestInteractive.java
  employees.txt
```



名稱	修改日期	類型	大小
.settings	2024/3/19 上午 11:46	檔案資料夾	
bin	2024/3/22 上午 09:36	檔案資料夾	
src	2024/3/19 上午 11:46	檔案資料夾	
.classpath	2023/10/26 下午 06:17	CLASSPATH 檔案	1 KB
.project	2023/10/26 下午 06:17	PROJECT 檔案	1 KB
employees.txt	2024/3/26 下午 02:09	文字文件	1 KB



```
1|Sean|Cheng|Mar 21, 1974|50000.00
2|Nicole|Wei|Nov 26, 1975|48000.00
3|David|Wang|Dec 25, 1980|40000.00
4|Ivy|Wu|Jun 15, 1984|32000.00
5|Louis|Chang|Feb 4, 1982|36000.00
```