CS5800 – Advanced Software Engineering

Cal Poly Pomona

Homework 1

Spring 2024

Description:

Inheritance

Polymorphism

Aggregation

Composition

Name: Fidelis Prasetyo

Email: (fprasetyo@cpp.edu)

BroncoID: 015765555

Github & Source code:

https://github.com/fidelisprasetyo/CS5800/tree/hw1

1. Inheritance

Source code:

- Employee.java

```
package inheritance;

public class Employee {
    private String firstName;
    private String lastName;
    private String socialSecurity;

    public Employee(String firstName, String lastName, String socialSecurity)
{
        this.firstName = firstName;
        this.lastName = lastName;
        this.socialSecurity = socialSecurity;
    }

    public void printEmployeeData() {
        System.out.println(
                "First name: " + getFirstName() + "\n" +
                "Last name: " + getLastName() + "\n" +
                "SSN: " + getSocialSecurity() + "\n");
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public void setSocialSecurity(String socialSecurity) {
        this.socialSecurity = socialSecurity;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getSocialSecurity() {
        return socialSecurity;
    }
}
```

- SalariedEmployee.java

```
package inheritance;

public class SalariedEmployee extends Employee {
    private int weeklySalary;

    public SalariedEmployee(String firstName, String lastName, String
socialSecurity, int weeklySalary) {
        super(firstName, lastName, socialSecurity);
        this.weeklySalary = weeklySalary;
    }

    @Override
    public void printEmployeeData() {
        System.out.println(
                "First name: " + getFirstName() + "\n" +
                "Last name: " + getLastName() + "\n" +
                "SSN: " + getSocialSecurity() + "\n" +
                "Weekly Salary: $" + weeklySalary + "\n");
    }

    public void setWeeklySalary(int weeklySalary) {
        this.weeklySalary = weeklySalary;
    }

    public int getWeeklySalary() {
        return weeklySalary;
    }
}
```

- HourlyEmployee.java

```
package inheritance;

public class HourlyEmployee extends Employee {
    private int wage;
    private int hoursWorked;

    public HourlyEmployee(String firstName, String lastName, String
socialSecurity, int wage, int hoursWorked) {
        super(firstName, lastName, socialSecurity);
        this.wage = wage;
        this.hoursWorked = hoursWorked;
    }

    @Override
    public void printEmployeeData() {
        System.out.println(
                "First name: " + getFirstName() + "\n" +
                "Last name: " + getLastName() + "\n" +
                "SSN: " + getSocialSecurity() + "\n" +
                "Wage: $" + wage + "\n" +
                "Hours worked: " + hoursWorked + "\n");
    }
```

```
    public void setWage(int wage) {
        this.wage = wage;
    }

    public void setHoursWorked(int hoursWorked) {
        this.hoursWorked = hoursWorked;
    }

    public int getWage() {
        return wage;
    }

    public int getHoursWorked() {
        return hoursWorked;
    }
}
```

- CommissionEmployee.java

```
package inheritance;

public class CommissionEmployee extends Employee {
    private int commissionRate;
    private int grossSales;

    public CommissionEmployee(String firstName, String lastName, String
socialSecurity, int commissionRate, int grossSales) {
        super(firstName, lastName, socialSecurity);
        this.commissionRate = commissionRate;
        this.grossSales = grossSales;
    }

    @Override
    public void printEmployeeData() {
        System.out.println(
                "First name: " + getFirstName() + "\n" +
                "Last name: " + getLastName() + "\n" +
                "SSN: " + getSocialSecurity() + "\n" +
                "Commission Rate: " + commissionRate + "%\n" +
                "Gross Salary: $" + grossSales + "\n");
    }

    public void setCommissionRate(int commissionRate) {
        this.commissionRate = commissionRate;
    }

    public void setGrossSales(int grossSales) {
        this.grossSales = grossSales;
    }

    public int getCommissionRate() {
        return commissionRate;
    }

    public int getGrossSales() {
        return grossSales;
```

```
    }
}
```

- BaseEmployee.java

```
package inheritance;

public class BaseEmployee extends Employee {
    private int baseSalary;

    public BaseEmployee(String firstName, String lastName, String
socialSecurity, int baseSalary) {
        super(firstName, lastName, socialSecurity);
        this.baseSalary = baseSalary;
    }

    @Override
    public void printEmployeeData() {
        System.out.println(
                "First name: " + getFirstName() + "\n" +
                "Last name: " + getLastName() + "\n" +
                "SSN: " + getSocialSecurity() + "\n" +
                "Base Salary: $" + baseSalary + "\n");
    }

    public void setBaseSalary(int baseSalary) {
        this.baseSalary = baseSalary;
    }

    public int getBaseSalary() {
        return baseSalary;
    }
}
```

- InheritanceDriver.java

```
package inheritance;

public class InheritanceDriver {

    public static void main(String[] args) {

        Employee joeJones = new SalariedEmployee("Joe", "Jones", "111-11-
1111", 2500);
        Employee stephSmith = new HourlyEmployee("Stephanie", "Smith", "222-
22-2222", 25, 32);
        Employee maryQuinn = new HourlyEmployee("Mary", "Quinn", "333-33-
3333", 19, 47);
        Employee nicoleDior = new CommissionEmployee("Nicole", "Dior", "444-
44-4444", 15, 50000);
        Employee renwaChanel = new SalariedEmployee("Renwa", "Chanel", "555-
55-5555", 1700);
        Employee mikeDavenport = new BaseEmployee("Mike", "Davenport", "666-
66-6666", 95000);
        Employee mahnazVaziri = new CommissionEmployee("Mahnaz", "Vaziri",
"777-77-7777", 22, 40000);
```

```
        joeJones.printEmployeeData();
        stephSmith.printEmployeeData();
        maryQuinn.printEmployeeData();
        nicoleDior.printEmployeeData();
        renwaChanel.printEmployeeData();
        mikeDavenport.printEmployeeData();
        mahnazVaziri.printEmployeeData();

    }
}
```

Output:

```
First name: Joe
Last name: Jones
SSN: 111-11-1111
Weekly Salary: $2500

First name: Stephanie
Last name: Smith
SSN: 222-22-2222
Wage: $25
Hours worked: 32

First name: Mary
Last name: Quinn
SSN: 333-33-3333
Wage: $19
Hours worked: 47

First name: Nicole
Last name: Dior
SSN: 444-44-4444
Commission Rate: 15%
Gross Salary: $50000

First name: Renwa
Last name: Chanel
SSN: 555-55-5555
Weekly Salary: $1700

First name: Mike
Last name: Davenport
SSN: 666-66-6666
Base Salary: $95000

First name: Mahnaz
Last name: Vaziri
SSN: 777-77-7777
Commission Rate: 22%
Gross Salary: $40000
```

2. Polymorphism

Source code:

- Ship.java

```
package polymorphism;

public class Ship {
    private String name;
    private String year;

    public Ship(String name, String year) {
        this.name = name;
        this.year = year;
    }

    public void print() {
        System.out.println(
                "Ship name: " + name + "\n" +
                "Build year: " + year + "\n");
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setYear(String year) {
        this.year = year;
    }

    public String getName() {
        return name;
    }

    public String getYear() {
        return year;
    }
}
```

- CruiseShip.java

```
package polymorphism;

public class CruiseShip extends Ship {
    private int maxPassengers;

    public CruiseShip(String name, String year, int maxPassengers) {
        super(name, year);
        this.maxPassengers = maxPassengers;
    }

    @Override
    public void print() {
        System.out.println(
```

```
                "Ship name: " + getName() + "\n" +
                "Maximum number of passengers: " + maxPassengers + "\n");
    }

    public void setMaxPassengers(int maxPassengers) {
        this.maxPassengers = maxPassengers;
    }

    public int getMaxPassengers() {
        return maxPassengers;
    }
}
```

- CargoShip.java

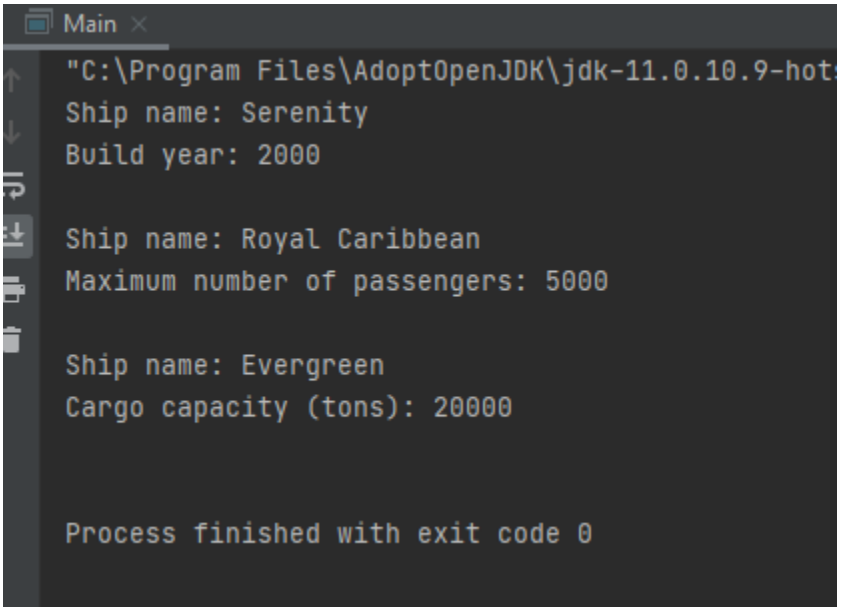
```
package polymorphism;

public class CargoShip extends Ship {
    private int cargoCapacity;

    public CargoShip(String name, String year, int cargoCapacity) {
        super(name, year);
        this.cargoCapacity = cargoCapacity;
    }

    @Override
    public void print() {
        System.out.println(
                "Ship name: " + getName() + "\n" +
                "Cargo capacity (tons): " + cargoCapacity + "\n");
    }

    public void setCargoCapacity(int cargoCapacity) {
        this.cargoCapacity = cargoCapacity;
    }

    public int getCargoCapacity() {
        return cargoCapacity;
    }
}
```

- PolymorphismDriver.java

```
package polymorphism;

public class PolymorphismDriver {

    public static void main(String[] args) {

        Ship[] shipArray = new Ship[3];

        shipArray[0] = new Ship("Serenity", "2000");
        shipArray[1] = new CruiseShip("Royal Caribbean", "2010", 5000);
        shipArray[2] = new CargoShip("Evergreen", "2020", 20000);
```

```
        for(Ship ship : shipArray) {
            ship.print();
        }
    }

}
```

Output:



3. Aggregation

Source code:

- Course.java

```
package aggregation;

import java.util.ArrayList;
import java.util.List;

public class Course {
    private String courseName;
    private List<Instructor> instructorList = new ArrayList<>();
    private List<Textbook> textbookList = new ArrayList<>();

    public Course(String courseName) {
        this.courseName = courseName;
    }

    public void print() {
        System.out.println("Course name: " + courseName);
```

```
        printInstructors();
        printTextbooks();
    }

    private void printInstructors() {
        System.out.println("Instructor(s): ");
        for(Instructor instructor : instructorList) {
            System.out.println("\t" + instructor.getFirstName() + " " +
instructor.getLastName());
        }
    }

    private void printTextbooks() {
        System.out.println("Textbook(s): ");
        for(Textbook textbook : textbookList) {
            System.out.println("\t" + "Title: " + textbook.getBookTitle());
            System.out.println("\t" + "Author: " + textbook.getAuthor());
        }
    }

    public void addInstructor(Instructor instructor) {
        instructorList.add(instructor);
    }

    public void addTextbook(Textbook textbook) {
        textbookList.add(textbook);
    }

    public void setCourseName(String courseName) {
        this.courseName = courseName;
    }

    public String getCourseName() {
        return courseName;
    }
}
```

- Instructor.java

```
package aggregation;

public class Instructor {
    private String firstName;
    private String lastName;
    private String officeNumber;

    public Instructor(String firstName, String lastName, String officeNumber)
{
        this.firstName = firstName;
        this.lastName = lastName;
        this.officeNumber = officeNumber;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
```

```
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public void setOfficeNumber(String officeNumber) {
        this.officeNumber = officeNumber;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public String getOfficeNumber() {
        return officeNumber;
    }
}
```

- Textbook.java

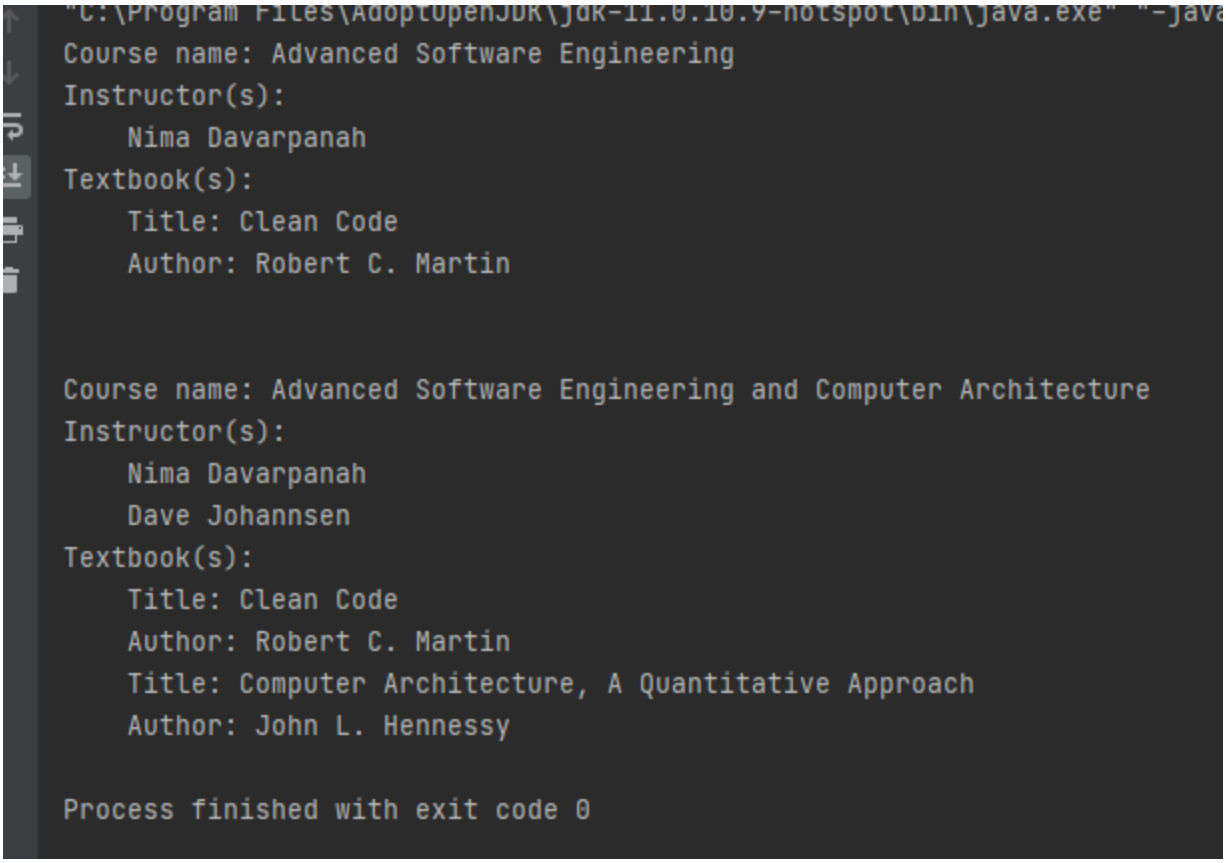
```
package aggregation;

public class Textbook {
    private String bookTitle;
    private String author;
    private String publisher;

    public Textbook(String bookTitle, String author, String publisher) {
        this.bookTitle = bookTitle;
        this.author = author;
        this.publisher = publisher;
    }

    public void setBookTitle(String bookTitle) {
        this.bookTitle = bookTitle;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }

    public String getBookTitle() {
        return bookTitle;
    }

    public String getAuthor() {
        return author;
    }
```

```
    public String getPublisher() {
        return publisher;
    }
}
```

- AggregationDriver.java

```
package aggregation;

public class AggregationDriver {

    public static void main(String[] args) {
        Instructor nimaDavarpanah = new Instructor("Nima", "Davarpanah", "3-
2636");
        Instructor daveJohannsen = new Instructor("Dave", "Johannsen", "1-
2345");

        Textbook cleanCode = new Textbook("Clean Code", "Robert C. Martin",
"Pearson");
        Textbook comArchitecture = new Textbook("Computer Architecture, A
Quantitative Approach", "John L. Hennessy", "Morgan Kaufmann");

        // First course with 1 instructor and 1 textbook
        Course advancedSwe = new Course("Advanced Software Engineering");
        advancedSwe.addInstructor(nimaDavarpanah);
        advancedSwe.addTextbook(cleanCode);

        // Second course with 2 instructors and 2 textbooks
        Course advancedSweCompArc = new Course("Advanced Software Engineering
and Computer Architecture");
        advancedSweCompArc.addInstructor(nimaDavarpanah);
        advancedSweCompArc.addInstructor(daveJohannsen);
        advancedSweCompArc.addTextbook(cleanCode);
        advancedSweCompArc.addTextbook(comArchitecture);

        advancedSwe.print();
        System.out.println("\n");
        advancedSweCompArc.print();

    }

}
```

Output:



4. Composition

Source code:

- Folder.java

```
package composition;

import java.util.ArrayList;
import java.util.List;

public class Folder {
    private String folderName;
    private List<Folder> folderList;
    private List<File> fileList;

    public Folder(String folderName) {
        this.folderName = folderName;
        this.folderList = new ArrayList<>();
        this.fileList = new ArrayList<>();
    }

    public void deleteFolder(Folder folder) {
```

```
        folderList.remove(folder);
    }

    public void deleteFile(File file) {
        fileList.remove(file);
    }

    public void print() {
        printHelper(0);
    }

    private void printHelper(int recurCount) {
        StringBuilder indentation = new StringBuilder();
        for(int i = 0; i < recurCount; i++) {
            indentation.append("\t");
        }

        System.out.println(indentation.toString() + "Folder : " +
folderName);

        for(Folder folder : folderList) {
            folder.printHelper(recurCount + 1);
        }

        for(File file : fileList) {
            System.out.println(indentation.toString() + "\tFile: " +
file.getFileName());
        }
    }

    public void addFolder(Folder folder) {
        folderList.add(folder);
    }

    public void addFile(File file) {
        fileList.add(file);
    }

    public void setFolderList(List<Folder> folderList) {
        this.folderList = folderList;
    }

    public void setFileList(List<File> fileList) {
        this.fileList = fileList;
    }

    public void setFolderName(String folderName) {
        this.folderName = folderName;
    }

    public List<Folder> getFolderList() {
        return folderList;
    }

    public List<File> getFileList() {
        return fileList;
    }
```

```
    public String getFolderName() {
        return folderName;
    }
}
```

- File.java

```
package composition;

public class File {
    private String fileName;

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

    public void setFileName(String fileName) {
        this.fileName = fileName;
    }

    public String getFileName() {
        return fileName;
    }

    public void print() {
        System.out.println("File: " + fileName);
    }
}
```

- CompositionDriver.java

```
package composition;

public class CompositionDriver {

    public static void main(String[] args) {

        Folder demo = new Folder("demo1");

        Folder sourceFiles = new Folder("Source Files");
        Folder includePath = new Folder("Include Path");
        Folder remoteFiles = new Folder("Remote Files");

        Folder phalcon = new Folder(".phalcon");
        Folder app = new Folder("app");

        Folder config = new Folder("config");
        Folder controllers = new Folder("controllers");
        Folder library = new Folder("library");
        Folder migrations = new Folder("migrations");
        Folder models = new Folder("models");
        Folder views = new Folder("views");

        Folder cache = new Folder("cache");
        Folder publicFolder = new Folder("public");
```

```
        File htaccess = new File(".htaccess");
        File htrouter = new File(".htrouter.php");
        File index = new File("index.html");

        demo.addFolder(sourceFiles);
        demo.addFolder(includePath);
        demo.addFolder(remoteFiles);

        sourceFiles.addFolder(phalcon);
        sourceFiles.addFolder(app);
        sourceFiles.addFolder(cache);
        sourceFiles.addFolder(publicFolder);
        sourceFiles.addFile(htaccess);
        sourceFiles.addFile(htrouter);
        sourceFiles.addFile(index);

        app.addFolder(config);
        app.addFolder(controllers);
        app.addFolder(library);
        app.addFolder(migrations);
        app.addFolder(models);
        app.addFolder(views);

        System.out.println(">> Full structure:");
        demo.print();

        // delete app folder
        System.out.println("\n>> Full structure after app folder deletion:");
        sourceFiles.deleteFolder(app);
        demo.print();

        // delete public folder
        System.out.println("\n>> Full structure after app & public folder
deletion:");
        sourceFiles.deleteFolder(publicFolder);
        demo.print();

    }
}
```

Output:

```
>> Full structure:
Folder : demo1
    Folder : Source Files
        Folder : .phalcon
        Folder : app
            Folder : config
            Folder : controllers
            Folder : library
            Folder : migrations
            Folder : models
            Folder : views
        Folder : cache
        Folder : public
        File: .htaccess
        File: .htrouter.php
        File: index.html
    Folder : Include Path
    Folder : Remote Files

>> Full structure after app folder deletion:
Folder : demo1
    Folder : Source Files
        Folder : .phalcon
        Folder : cache
        Folder : public
        File: .htaccess
        File: .htrouter.php
        File: index.html
    Folder : Include Path
    Folder : Remote Files

>> Full structure after app & public folder deletion:
Folder : demo1
    Folder : Source Files
        Folder : .phalcon
        Folder : cache
        File: .htaccess
        File: .htrouter.php
        File: index.html
    Folder : Include Path
    Folder : Remote Files

Process finished with exit code 0
```

Source Code & Supporting Files

The entire source code and other supporting documents/ files can be obtained from this GitHub repository:

https://github.com/fidelisprasetyo/CS5800/tree/hw1