Daniel Plascencia   
Repository Link: <https://github.com/ExodiusLux/CS-4800/tree/main/Homework_3>

**Homework 2**

**UML**

**NOTE:**

Use Solution Provided in Class

You must use **PlantUML**

**Class Diagram**

Your assignment is to go back to **Homework 1** and create a UML Class Diagram for all parts.

Part 1,2,3 and 4.

Part 1 (homework\_3\_1\_pt1.puml)

@startuml  
abstract class Employee{  
 -String first\_name  
 -String last\_name  
 -String SSN  
  
 +getfirstName() : String  
 +getlastName() : String  
 +getSSN() : String  
 +setfirstName(String first\_name)  
 +setlastName(String last\_name)  
 +setSSN(String SSN)  
}  
class SalariedEmployee extends Employee{  
 -int weekly\_salary  
  
 +getweeklySalary() : int  
 +setweeklySalary(int weekly\_salary)  
}  
class HourlyEmployee extends Employee{  
 -int wage  
 -int hours\_worked  
  
 +getWage() : int  
 +gethoursWorked() : int  
 +setWage(int wage)  
 +sethoursWorked(int hours\_worked)  
}  
class CommissionBasedEmployee extends Employee{  
 -double commission\_rate  
 -int gross\_sales  
  
  
 +getcommissionRate() : double  
 +get grossSales(): int  
 +setcommissionRate(double commission\_rate)  
 +setgrossSales(int gross\_sales)  
}  
class BaseEmployee extends Employee{  
 -int base\_salary  
  
 +getbaseSalary() : int  
 +setbaseSalart(int base\_salary)  
}  
  
@enduml

A screenshot of a computer

Description automatically generated

Part 2

@startuml  
  
class Ship{  
 -String ship\_name  
 -String year\_built  
  
 +getShipName() : String  
 +getYearBuilt() : String  
 +setShipName(String ship\_name)  
 +setYearBuilt(String year\_built)  
 +print()  
}  
class CruiseShip extends Ship{  
 -int num\_of\_Passengers  
  
 +getNumofPassengers() : int  
 +setNumofPassengers(int num\_of\_Passengers)  
 +print()  
}  
class CargoShip extends Ship{  
 -int cargo\_capacity  
  
 +getCargoCapacity() : int  
 +setCargoCapacity(int cargo\_capacity)  
 +print()  
  
}  
  
@enduml

A screenshot of a computer

Description automatically generated

Part 3

@startuml  
class Course {  
 Instructor instructor  
 Textbook textbook  
 String course\_name  
  
 +getInstructor() : Instructor  
 +getTextbook() : Textbook  
 +getCourseName() : String  
 +setInstructor(Instructor instructor)  
 +setTextbook(Textbook textbook)  
 +setCourseName(String course\_name)  
 +print()  
  
}  
class Instructor {  
 -String first\_name  
 -String last\_name  
 -String office\_number  
  
 +getFirstName() : String  
 +getLastName() : String  
 +getOfficeNumber() : String  
 +setFirstName(String first\_name)  
 +setLastName(String last\_name)  
 +setOfficeNumber(String office\_number)  
}  
class Textbook {  
 -String title  
 -String author  
 -String publisher  
  
 +getTitle() : String  
 +getAuthor() : String  
 +getPublisher() : String  
 +setTitle(String title)  
 +setAuthor(String author)  
 +setPublisher(String publisher)  
  
}  
  
Course "1..\*" o-- "0..\*"Instructor  
Course "1..\*" o-- "0..\*" Textbook  
  
@enduml

A diagram of a course

Description automatically generated

Part 4

@startuml  
class File{  
 -String file\_name  
  
 +getFileName(): String  
 +setFileName(String file\_name)  
 +print()  
}  
class Folder{  
 -Object[] files  
 -Object[] subfolders  
 String subfolder\_name  
  
  
 +getFiles(): Object[]  
 +getFolders(): Object []  
 +getFolderName() : String  
 +setFolderName(String folder\_name)  
 +removesubFolder(String subfolder\_name)  
 +addsubFolder(String subfolder\_name)  
 +getsubFolder(): Folder  
 +removeFile(File file)  
 +addFile(String file\_name)  
 +print()  
 +printFolderContents()  
}  
@enduml

A screenshot of a computer

Description automatically generated

**Object Diagram**

Create the UML **Object Diagram** for the following parts in HW 1 at specific points specified.

1) **Part 2** after all 3 ships have been dynamically created.

@startuml  
object "Admire:Ship" as sh{  
 ship\_name = "Admire"  
 year\_built = "1997"  
}  
object "Carnival Cruise:Cruiseship" as carnival{  
 ship\_name = "Carnival Cruise"  
 year\_built = 2019  
 num\_of\_passengers = 3000  
}  
  
object "Tankers:Cargoship" as tank{  
 ship\_name = "Tankers"  
 year\_built = "2011"  
 cargo\_capacity = 200000  
}  
  
@enduml

A group of text boxes

Description automatically generated

2) **Part 3** after step 5. Before adding a second instructor and book.

@startuml  
object "Admire:Ship" as sh{  
 ship\_name = "Admire"  
 year\_built = "1997"  
}  
object "Carnival Cruise:Cruiseship" as carnival{  
 ship\_name = "Carnival Cruise"  
 year\_built = 2019  
 num\_of\_passengers = 3000  
}  
  
object "Tankers:Cargoship" as tank{  
 ship\_name = "Tankers"  
 year\_built = "2011"  
 cargo\_capacity = 200000  
}  
  
@enduml

A diagram of a course

Description automatically generated

3) **Part 4** after step 2 when the app folder is deleted.

@startuml  
  
object "php\_demo1:Folder" as php {  
 subfolder\_name = php\_demo1  
}  
object "Source Files:Folder" as src{  
 subfolder\_name = "Source Files"  
}  
object ".phalcon:Folder" as phal {  
 subfolder\_name = .phalcon  
}  
object "cache:Folder" as ca {  
 subfolder\_name = cache  
}  
object "public:Folder" as pub{  
 subfolder\_name = public  
}  
object ".htaccess:File" as hta {  
 file\_name = .htaccess  
}  
object ".htrouter.php:File" as htr{  
 file\_name = .htrouter.php  
}  
object "index.html:File" as ind {  
 file\_name = index.html  
}  
  
php \*-- src  
  
src \*-- phal  
src \*-- ca  
src \*-- pub  
  
pub \*-- hta  
pub \*-- htr  
pub \*-- ind  
@enduml

A diagram of a computer

Description automatically generated