Lab 13

Requirements:

- Create a Java project named yourStudentId_OOP_Lab13
- Read instructions and create classes needed. You are supposed to create 1 interface and 4 classes (FixedAsset, Vehicle, Machine, Company, Tester).
- All instance variables are private. Please use public methods to access private instance variables.

Description:

In the accounting field, there are many methods for calculating the depreciation of fixed assets, such as the straight-line method and the double-declining balance method. We can use these formulas and some information like the cost, salvage value, salvage value, or service life of fixed assets to calculate depreciation expenses. In order to make the balance sheet clearly show the company's financial status, in addition to respond to the principle of honesty, investors can also understand the company's status better. Now, you are ordered to develop a system for calculating depreciation expenses and the book value of fixed assets. This question assumes that the company currently only has two major assets, Vehicle, and Machine, and they use double declining balance depreciation method and straight-line method to depreciate, respectively. In addition, using the GUI to display the results allows the manager to get more information. The UML of the exercises is shown in Figure 1.

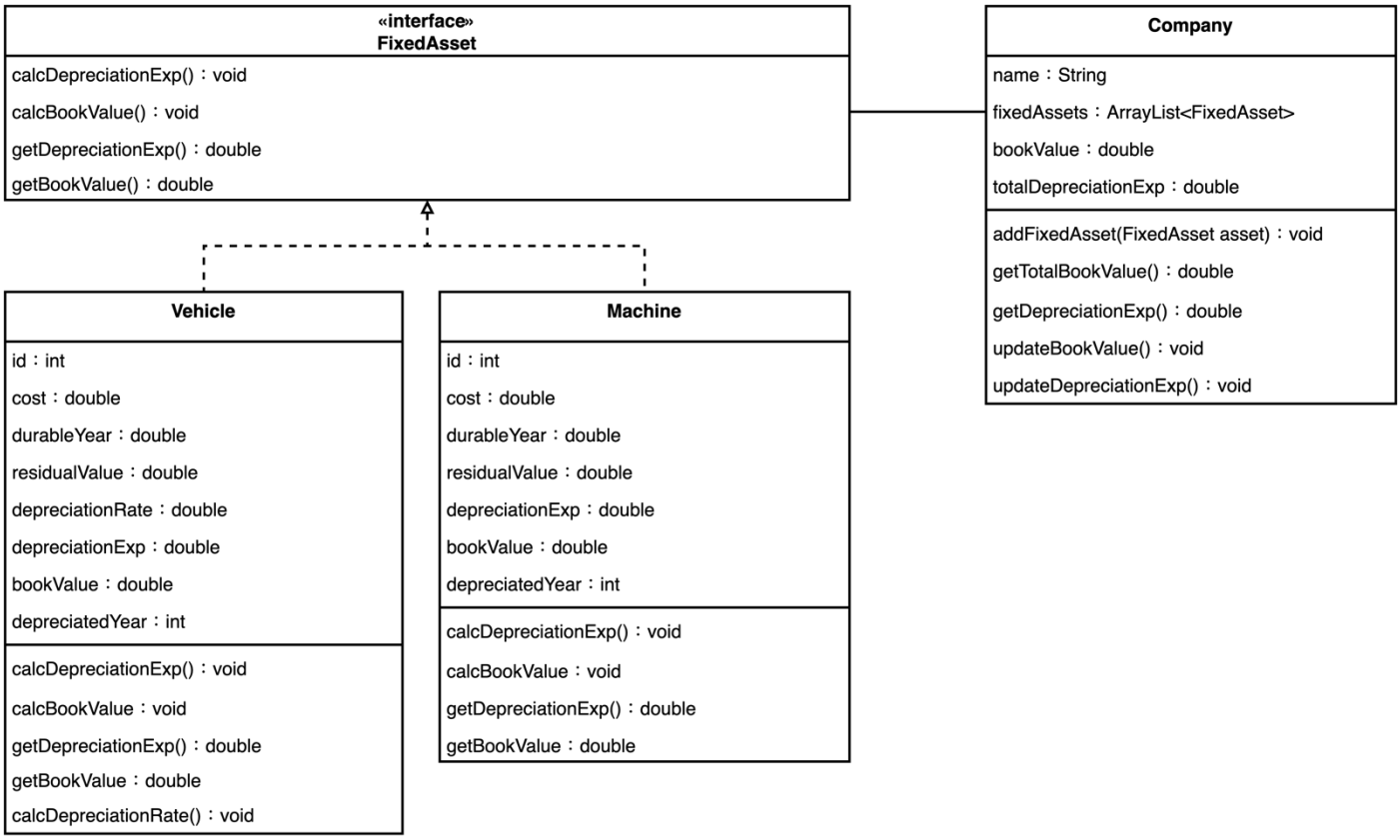


Figure 1. The UML diagram of the exercise problem

1. **Create *FixedAsset* interface**

FixedAsset	
Modifier and type	**Method (or Variable) and description**
Abstract methods	
void	calcDepreciationExp() The abstract method is used to calculate the depreciation expense of the fixed assets.
void	calcBookValue() The abstract method is used to calculate the book value.
double	getDepreciationExp() The abstract method is used to return the depreciation expense.
double	getBookValue() The abstract method is used to return the book value.

2. **Create *Vehicle* class**

Vehicle	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	id The identity of the vehicle.
double	cost The original cost of the vehicle.
double	durableYear The expected duration of year which vehicles can be used.
double	residualValue The amount that a company expects to receive on the end of the asset service life.
double	depreciationRate Annual depreciation rate of double-declining balance method.
double	depreciationExp Annual depreciation expense of the fixed asset.
double	bookValue Current book value of the fixed asset.
int	depreciationYear Record of the number of depreciated periods.
Constructor	
Vehicle(int id, double cost, double durableYear, double residualValue) Enable to instantiate the object of *Vehicle* with given id, cost, durable year, residual value, and book value. Moreover, initialize the value of depreciated year as 0 and *calcDepreciationRate()*.	
Instance methods	

void	calcDepreciationRate() Calculate the depreciation rate with the formula below: ***depreciation rate = (1 / durable year) * 2***
void	calcDepreciationExp() Three conditions to calculate the depreciation expense: 1. When the durable year is bigger than the depreciated year, calculate with the formula below and update the depreciated year and book value: ***depreciation expense = book value * depreciation rate*** 2. When the durable year is the same as the depreciated year added one, calculate with the formula below and update the depreciated year and book value: ***depreciation expense = book value – residual value*** 3. Otherwise, set the depreciation expense as zero.
void	calcBookValue() Update and calculate the book value with depreciation expense.
double	getDepreciationExp() Call the method to calculate the depreciation expense and return it.
double	getBookValue() Call the method to calculate the book value and return the it.

3. **Create *Machine* class**

Machine	
Modifier and type	**Method (or Variable) and description**
Instance variable	
int	id The identity of the machine.
double	cost The cost of the machine.
double	durableYear The duration of year which machine can be used.
double	residualValue The amount that a company expects to receive for an asset at the end of its service life.
double	depreciationExp Annual depreciation expense of the fixed asset.
double	bookValue Current book value of the fixed asset.
int	depreciationYear Record of the number of depreciated periods.
Constructor	

Machine(int id, double cost, double durableYear, double residualValue) Enable to instantiate the object of *Machine* with given id, cost, durable year, and residual value. Moreover, initialize the depreciated year as 0.	
Instance methods	
void	calcDepreciationExp() Two conditions to calculate the depreciation expense: 1. When the durable year is bigger than the depreciated year, calculate with the formula below and update the depreciated year and book value: ***depreciation expense = (cost – residual value) / durable year*** 2. Otherwise, set the depreciation expense as zero.
void	calcBookValue() Update and calculate the book value with depreciation expense.
double	getDepreciationExp() Call the corresponding method to get the depreciation expense and return it.
double	getBookValue() Call the corresponding method to get the book value and return it.

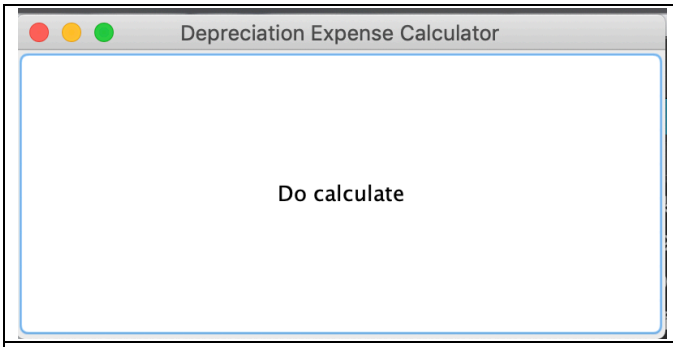
4. **Create *Company* class**

Company	
Modifier and type	**Method (or Variable) and description**
Instance variable	
String	name The company's name
ArrayList<FixedAsset>	fixedAssets An *ArrayList* which is used to store the fixed assets.
double	bookValue The sum of the book value of fixed assets in the current year.
double	totalDepreciationExp The sum of depreciation expense of the company in the current year.
Constructor	
Company(String name) Enable to instantiate the object of *Company* with given name, and initialize the ArrayList. Moreover, initialize both the book value and total depreciation expense as 0.	
Instance methods	
void	addFixedAsset(FixedAsset asset) Add all the assets into the fixed assets ArrayList and add the book value of the asset to the current book value.
double	getTotalBookValue()

	Call the corresponding method and return the current year's book value.
double	getDepreciationExp() Call the corresponding method and return the current year's depreciation expense.
void	updateBookValue Update and calculate the book value with the total depreciation expense.
void	updateDepreciationExp() Update and calculate all the total depreciation expenses of the fixed assets.

5. **Create *Tester* class**
   - Import the needed package.
   - Define the frame size with final static constant named ***FRAME_WIDTH*** and ***FRAME_HEIGHT*** and assign them with the related value 400 and 200.
   - Instantiate the *Company* with its' name *NCCU* and add the fixed assets *Vehicle* with the *id* as1, *cost* as 500000, *durableYear* as 10, and *residualValue* as 50000.
   - Create the components with *JFrame*, and *JButton* displayed in GUI and shown the result as the sample output on the console.
   - Using *inner-class* concept to implement the *ActionListener* interface, and initialize the private variable *currentYear* as 0.
   - Overriding the *actionPerformed(ActionEvent e)* method to print the string of current year, the company's depreciation expense, and the total book value as the sample output.
   - Initializing the *ActionListener* object named *button* with *ButtonListener* constructor.
   - Add buttonListener to the *button.*
   - Set the frame *size* as ***FRAME_WIDTH*** and ***FRAME_HEIGHT,*** default close operation as JFrame.EXIT_ON_CLOSE, and the frame visible as true.

Sample output

	Year: 1 Total depreciation expense: 100000.00 Book value of fixed assets : 400000.00 Year: 2 Total depreciation expense: 80000.00 Book value of fixed assets : 320000.00
The appearance of GUI	The output on the console

Submission: Submit your project as ".zip file" via Moodle. No other submissions will be graded.

Reminder: Please zip the whole project

Deadline: Tomorrow's midnight (for both Mon56 and Tue23)