|  |
| --- |
| ExpenseItem |
| -name:String  -amount:double  -times:int |
| +<<constructor>> ExpenseItem(name:String, amount:double, times:int)  +getName():String  +setName(name:String):void  +getAmount():double  +setAmount(amount:double):void  +getTimes():int  +setTimes(times:int):void  +annualTotal():double  +getFrequncy():String  +toString():String |

ExpenseItem source code (For the report)

**public** **class** ExpenseItem {

**private** String name; //3 private instance variables

**private** **double** amount;

**private** **int** times;

**public** ExpenseItem(String name, **double** amount, **int** times) {

**this**.name = name;

**this**.amount = amount;

**this**.times = times;

}//constructor class

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getAmount() {

**return** amount;

}

**public** **void** setAmount(**double** amount) {

**this**.amount = amount;

}

**public** **int** getTimes() {

**return** times;

}

**public** **void** setTimes(**int** times) {

**this**.times = times;

}

**public** **double** annualTotal() {

**double** total = amount\*times;

**return** total; //returns the totals for each item

}

**public** String getFrequency() {

**if** (times == 12)

**return** "Monthly";//payments are made once a month

**else** **if** (times == 52)

**return** "Weekly";//payments are made once a week

**else** **if** (times == 1)

**return** "Yearly";//payment is made once a year

**else** **if** (times == 2)

**return** "Semi-Annually";//payment is made twice a year

**else** **if** (times == 182)

**return** "Every Other Day";//payment is made every other day since 182\*2=364 days

**else**

**return** "Times per year";//payment is made (n) times per year

}

**public** String toString() {

**return** String.*format*("%-10s\t%10.2f\t%10d\t%10s\t%15.2f", name, amount, times, getFrequency(), annualTotal());//format for body output, method is called in ExpenseEstimator

}//3 private instance variables, 2 methods, make up 5 columns of data for toString() for output

}

ExpenseEstimator Source Code (For report)

**public** **class** ExpenseEstimator {

**public** **static** **void** main(String[] args) {

ExpenseItem[] items = **new** ExpenseItem[9]; //each item defined for its name. amount, and frequency per year

items[0] = **new** ExpenseItem("Coffee", 2.67, 182);

items[1] = **new** ExpenseItem("Food", 8.50, 260);

items[2] = **new** ExpenseItem("School Parking", 450.00, 1);

items[3] = **new** ExpenseItem("Phone", 38.90, 12);

items[4] = **new** ExpenseItem("Tuition", 8550.47, 2);

items[5] = **new** ExpenseItem("Gas", 48.50, 52);

items[6] = **new** ExpenseItem("Entertainment", 60.00, 1);

items[7] = **new** ExpenseItem("Books", 100.00, 2);

items[8] = **new** ExpenseItem("Car Insurance", 170.00, 8);

**double** grandtotal = 0;//declare and initialize grand total to 0

System.***out***.printf("%-10s\t%10s\t%10s\t%10s\t%15s\n", "Item", "Amount", "Frequency", "Basis", "Annual Total"); //print header

**for** (**int** i=0; i<items.length; i++) { //begin for loop

**double** annualtotal = items[i].annualTotal(); //declare annual total

grandtotal += annualtotal; //add annual total to grand total

System.***out***.println(items[i].toString()); //call toString() from ExpenseItem with format

}//end for loop

System.***out***.printf("%-10s\t%10s\t%10s\t%10s\t%15.2f", "GrandTotal", "", "", "", grandtotal); //grand total output

}

}