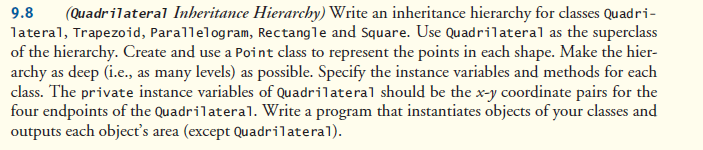
Lab8 CIS43 Due: 7/12/2016

Name: Nikhil Vytla

***Exercise: Ex 9.8 and 9.14***



Point

**package** P98;

**public** **class** Point {

**private** **int** x;

**private** **int** y;

**public** Point(**int** x, **int** y)

{

//this.x = x;

//this.y = y;

}

**public** **int** getX() {

**return** x;

}

**public** **void** setX(**int** x) {

**this**.x = x;

}

**public** **int** getY() {

**return** y;

}

**public** **void** setY(**int** y) {

**this**.y = y;

}

}

-------------------------------------------------

Quadrilateral (Superclass)

**package** P98;

**public** **abstract** **class** Quadrilateral

{

**public** Point topleft = **new** Point(0,0);

**public** Point topright = **new** Point(0,0);

**public** Point bottomright = **new** Point(0,0);

**public** Point bottomleft = **new** Point(0,0);

**public** **abstract** **double** getArea();

}

-------------------------------------------------

Trapezoid

**public** **class** Trapezoid **extends** Quadrilateral {

**public** Trapezoid(**int** x1, **int** y1, **int** x2, **int** y2, **int** x3, **int** y3, **int** x4, **int** y4)

{

topleft.setX(x1);

topleft.setY(y1);

topright.setX(x2);

topright.setY(y2);

bottomright.setX(x3);

bottomright.setY(y3);

bottomleft.setX(x4);

bottomleft.setY(y4);

}

**int** top\_length;

**int** bottom\_length;

**int** right\_height;

**int** left\_height;

**public** **double** getArea()

{

top\_length = topright.getX() - topleft.getX();

bottom\_length = bottomright.getX() - bottomleft.getX();

right\_height = topright.getY() - bottomright.getY();

left\_height = topleft.getY() - bottomleft.getY();

System.*out*.printf("Trapezoid Trap1's dimensions: %nBase1: %d%nBase2: %d%nHeight: %d", top\_length, bottom\_length, right\_height);

**double** area = (0.5 \* (top\_length + bottom\_length) \* right\_height);

**return** area;

}

}

-------------------------------------------------

Parallelogram

**package** P98;

**public** **class** Parallelogram **extends** Trapezoid{

**public** Parallelogram(**int** x1, **int** y1, **int** x2, **int** y2, **int** x3, **int** y3,**int** x4, **int** y4)

{

**super**(x1, y1, x2, y2, x3, y3, x4, y4);

}

**public** **double** getArea()

{

bottom\_length = bottomright.getX() - bottomleft.getX();

right\_height = topright.getY() - bottomright.getY();

System.*out*.printf("%nParallelogram Par1's dimensions: %nBase: %d %nHeight: %d", bottom\_length, right\_height);

**double** area = (bottom\_length \* right\_height);

**return** area;

}

}

-------------------------------------------------

Rectangle

**package** P98;

**public** **class** Rectangle **extends** Parallelogram{

**public** Rectangle(**int** x1, **int** y1, **int** x2, **int** y2, **int** x3, **int** y3, **int** x4, **int** y4) {

**super**(x1, y1, x2, y2, x3, y3, x4, y4);

}

**public** **double** getArea()

{

bottom\_length = bottomright.getX() - bottomleft.getX();

right\_height = topright.getY() - bottomright.getY();

System.*out*.printf("%nRectangle Rect1's dimensions: %nBase: %d %nHeight: %d", bottom\_length, right\_height);

**double** area = (bottom\_length \* right\_height);

**return** area;

}

}

-------------------------------------------------

Square

**package** P98;

**public** **class** Square **extends** Rectangle{

**public** Square(**int** x1, **int** y1, **int** x2, **int** y2, **int** x3, **int** y3, **int** x4, **int** y4) {

**super**(x1, y1, x2, y2, x3, y3, x4, y4);

}

**public** **double** getArea()

{

top\_length = topright.getX() - topleft.getX();

bottom\_length = bottomright.getX() - bottomleft.getX();

System.*out*.printf("%nParallelogram Square1's dimensions: %nBase: %d %nHeight: %d", top\_length, bottom\_length);

**double** area = (top\_length \* bottom\_length);

**return** area;

}

}

-------------------------------------------------

QuadrilateralTest

**package** P98;

**public** **class** QuadrilateralTest {

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

Trapezoid Trap1 = **new** Trapezoid(1,4,3,4,4,0,0,0);

Parallelogram Par1 = **new** Parallelogram(1,2,6,2,5,0,0,0);

Rectangle Rect1 = **new** Rectangle(0,2,6,2,6,0,0,0);

Square Square1 = **new** Square(0,7,7,7,7,0,0,0);

System.*out*.printf("%nTrap1's area: %.2f%n", Trap1.getArea());

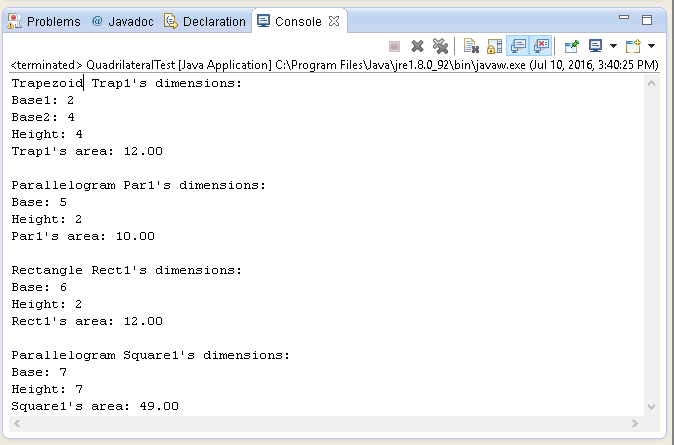
System.*out*.printf("%nPar1's area: %.2f%n", Par1.getArea());

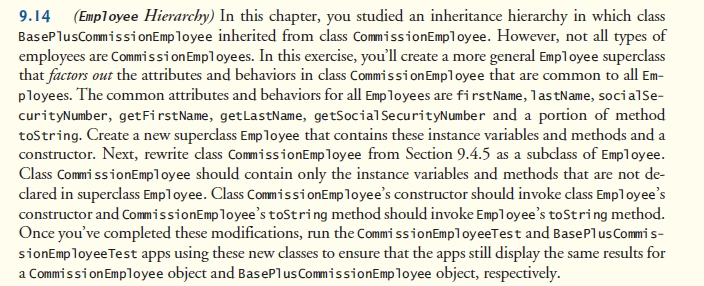
System.*out*.printf("%nRect1's area: %.2f%n", Rect1.getArea());

System.*out*.printf("%nSquare1's area: %.2f", Square1.getArea());

}

}





Employee (Superclass)

**package** P914;

**public** **class** Employee {

**private** **final** String firstName;

**private** **final** String lastName;

**private** **final** String socialSecurityNumber;

**public** Employee(String firstName, String lastName, String socialSecurityNumber)

{

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.socialSecurityNumber = socialSecurityNumber;

}

**public** String getFirstName()

{

**return** firstName;

}

**public** String getLastName()

{

**return** lastName;

}

**public** String getSocialSecurityNumber()

{

**return** socialSecurityNumber;

}

**public** String toString()

{

**return** String.*format*("%s: %s %s%n%s: %s%n",

"commission employee", getFirstName(), getLastName(),

"social security number", getSocialSecurityNumber());

}

}

-------------------------------------------------

CommissionEmployee

**package** P914;

//Fig. 9.10: CommissionEmployee.java

//CommissionEmployee class uses methods to manipulate its

//private instance variables.

**public** **class** CommissionEmployee **extends** Employee

{

**private** **double** grossSales; // gross weekly sales

**private** **double** commissionRate; // commission percentage

// two-argument constructor

**public** CommissionEmployee(String firstName, String lastName, String socialSecurityNumber, **double** grossSales, **double** commissionRate)

{

**super**(firstName, lastName, socialSecurityNumber);

// implicit call to Object constructor occurs here

// if grossSales is invalid throw exception

**if** (grossSales < 0.0)

**throw** **new** IllegalArgumentException(

"Gross sales must be >= 0.0");

// if commissionRate is invalid throw exception

**if** (commissionRate <= 0.0 || commissionRate >= 1.0)

**throw** **new** IllegalArgumentException(

"Commission rate must be > 0.0 and < 1.0");

**this**.grossSales = grossSales;

**this**.commissionRate = commissionRate;

} // end constructor

// set gross sales amount

**public** **void** setGrossSales(**double** grossSales)

{

**if** (grossSales < 0.0)

**throw** **new** IllegalArgumentException(

"Gross sales must be >= 0.0");

**this**.grossSales = grossSales;

}

// return gross sales amount

**public** **double** getGrossSales()

{

**return** grossSales;

}

// set commission rate

**public** **void** setCommissionRate(**double** commissionRate)

{

**if** (commissionRate <= 0.0 || commissionRate >= 1.0)

**throw** **new** IllegalArgumentException(

"Commission rate must be > 0.0 and < 1.0");

**this**.commissionRate = commissionRate;

}

// return commission rate

**public** **double** getCommissionRate()

{

**return** commissionRate;

}

// calculate earnings

**public** **double** earnings()

{

**return** getCommissionRate() \* getGrossSales();

}

// return String representation of CommissionEmployee object

@Override

**public** String toString()

{

**return** String.*format*("%s%n%s: %.2f%n%s: %.2f",

**super**.toString(),

"gross sales", getGrossSales(),

"commission rate", getCommissionRate());

}

} // end class CommissionEmployee

-------------------------------------------------

BasePlusCommissionEmployee

**package** P914;

// Fig. 9.11: BasePlusCommissionEmployee.java

// BasePlusCommissionEmployee class inherits from CommissionEmployee

// and accesses the superclass’s private data via inherited

// public methods.

**public** **class** BasePlusCommissionEmployee **extends** CommissionEmployee

{

**private** **double** baseSalary; // base salary per week

// six-argument constructor

**public** BasePlusCommissionEmployee(String firstName, String lastName,

String socialSecurityNumber, **double** grossSales,

**double** commissionRate, **double** baseSalary)

{

**super**(firstName, lastName, socialSecurityNumber,

grossSales, commissionRate);

// if baseSalary is invalid throw exception

**if** (baseSalary < 0.0)

**throw** **new** IllegalArgumentException(

"Base salary must be >= 0.0");

**this**.baseSalary = baseSalary;

}

// set base salary

**public** **void** setBaseSalary(**double** baseSalary)

{

**if** (baseSalary < 0.0)

**throw** **new** IllegalArgumentException(

"Base salary must be >= 0.0");

**this**.baseSalary = baseSalary;

}

// return base salary

**public** **double** getBaseSalary()

{

**return** baseSalary;

}

// calculate earnings

@Override

**public** **double** earnings()

{

**return** getBaseSalary() + **super**.earnings();

}

// return String representation of BasePlusCommissionEmployee

@Override

**public** String toString()

{

**return** String.*format*("%s %s%n%s: %.2f", "base-salaried",

**super**.toString(), "base salary", getBaseSalary());

}

} // end class BasePlusCommissionEmployee

-------------------------------------------------

CommissionEmployeeTest

**package** P914;

// Fig. 9.5: CommissionEmployeeTest.java

// CommissionEmployee class test program.

**public** **class** CommissionEmployeeTest

{

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

{

// instantiate CommissionEmployee object

CommissionEmployee employee = **new** CommissionEmployee(

"Sue", "Jones", "222-22-2222", 10000, .06);

// get commission employee data

System.*out*.println(

"Employee information obtained by get methods:");

System.*out*.printf("%n%s %s%n", "First name is",

employee.getFirstName());

System.*out*.printf("%s %s%n", "Last name is",

employee.getLastName());

System.*out*.printf("%s %s%n", "Social security number is",

employee.getSocialSecurityNumber());

System.*out*.printf("%s %.2f%n", "Gross sales is",

employee.getGrossSales());

System.*out*.printf("%s %.2f%n", "Commission rate is",

employee.getCommissionRate());

employee.setGrossSales(5000);

employee.setCommissionRate(.1);

System.*out*.printf("%n%s:%n%n%s%n",

"Updated employee information obtained by toString", employee);

} // end main

} // end class CommissionEmployeeTest

-------------------------------------------------

BasePlusCommissionEmployeeTest

**package** P914;

// BasePlusCommissionEmployeeTest.java

// Testing class BasePlusCommissionEmployee.

**public** **class** BasePlusCommissionEmployeeTest

{

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

{

// instantiate BasePlusCommissionEmployee object

BasePlusCommissionEmployee employee =

**new** BasePlusCommissionEmployee(

"Bob", "Lewis", "333-33-3333", 5000, .04, 300);

// get base-salaried commission employee data

System.*out*.println(

"Employee information obtained by get methods:");

System.*out*.printf("%n%s %s%n", "First name is",

employee.getFirstName());

System.*out*.printf("%s %s%n", "Last name is",

employee.getLastName());

System.*out*.printf("%s %s%n", "Social security number is",

employee.getSocialSecurityNumber());

System.*out*.printf("%s %.2f%n", "Gross sales is",

employee.getGrossSales());

System.*out*.printf("%s %.2f%n", "Commission rate is",

employee.getCommissionRate());

System.*out*.printf("%s %.2f%n", "Base salary is",

employee.getBaseSalary());

employee.setBaseSalary(1000);

System.*out*.printf("%n%s:%n%n%s%n",

"Updated employee information obtained by toString",

employee.toString());

} // end main

} // end class BasePlusCommissionEmployeeTest

