# 1 užduotis:

## Main:

public class part5Main {  
 public static Double sumOfAreas(ArrayList<Triangle> triangles){  
 Double sum = 0.0;  
 for (Triangle shape:triangles) {  
 sum+=shape.area();  
 }  
 return sum;  
 }  
 public static void main(String[] args) {  
 ArrayList <Triangle> triangles = new ArrayList<Triangle>();  
  
 Triangle triangle = new Triangle(5.0,5.0, 5.0);  
 triangles.add(triangle);  
 RightTriangle triangle2 = new RightTriangle(5.0,5.0);  
 triangles.add(triangle2);  
 EquilateralTriangle triangle3 = new EquilateralTriangle(5.0);  
 triangles.add(triangle3);  
  
 for (Triangle shape:triangles) {  
 System.*out*.println(shape.area());  
 }  
  
 System.*out*.println("Bendra plotų suma suapvalinus: " + Math.*round*(*sumOfAreas*(triangles)));  
  
 }  
}

## Triangle:

public class Triangle {  
 Double edgeFirst;  
 Double edgeSecond;  
 Double edgeThird;  
  
 Triangle(){}  
  
 Triangle(Double \_edgeFirst, Double \_edgeSecond, Double \_edgeThird){  
 edgeFirst = \_edgeFirst;  
 edgeSecond = \_edgeSecond;  
 edgeThird = \_edgeThird;  
 }  
  
 public Double area(){  
 Double halfPerimeter = (edgeFirst + edgeSecond + edgeThird)/2.0;  
 return Math.*sqrt*(halfPerimeter\*(halfPerimeter-edgeFirst)\*(halfPerimeter-edgeSecond)\*(halfPerimeter-edgeThird));  
 }  
}

## RightTriangle (Statusis):

public class RightTriangle extends Triangle{  
  
 RightTriangle(Double \_edgeFirst, Double \_edgeSecond){  
 edgeFirst = \_edgeFirst;  
 edgeSecond = \_edgeSecond;  
 }  
  
 public void findEdgeThird(){  
 edgeThird = Math.*sqrt*(edgeFirst \* edgeFirst + edgeSecond \* edgeSecond);  
 }  
  
 public Double area() {  
 return (edgeFirst \* edgeSecond) / 2.0;  
 }  
}

## EquilateralTriangle (lygiakraštis):

public class EquilateralTriangle extends Triangle{  
 EquilateralTriangle(Double \_edgeFirst){  
 edgeFirst = \_edgeFirst;  
 }  
  
 public void edgesValues(){  
 edgeSecond = edgeFirst;  
 edgeThird = edgeFirst;  
 }  
  
 public Double area(){  
 return (edgeFirst \* edgeFirst \* Math.*sin*(Math.*toRadians*(60)))/2.0;  
 }  
}

# 2 užduotis:

## Main:

public class part6Main {  
 public static void main(String[] args) {  
 Staff company = new Staff();  
  
 Employee empl1 = new Employee("Jonas", "Jonaitis", "861515515",1234, 1000);  
 company.addStaffMember(empl1);  
  
 Employee empl2 = new Employee("Petras", "Petraitis", "861512315",1235, 900);  
 company.addStaffMember(empl2);  
  
 Trainee tr = new Trainee("Lukas", "Lukaitis", "861123315");  
 company.addStaffMember(tr);  
  
 Executive exe = new Executive("Marytė", "Marytytė", "861233315", 1236, 1100, 150);  
 company.addStaffMember(exe);  
  
 Hourly hour1 = new Hourly("Rimas", "Rimaitis","861233515", 1237, 10, 5.5);  
 company.addStaffMember(hour1);  
  
 Hourly hour2 = new Hourly("Ona", "Onaitė","861235925", 1238, 20, 5.5);  
 company.addStaffMember(hour2);  
  
 System.*out*.println(company.payDay());  
  
 }  
}

## Staff:

public class Staff {  
 StaffMember[] staffList;  
   
 Staff(){  
 staffList = new StaffMember[10];  
 }  
  
 void addStaffMember(StaffMember staff){  
 for (int i=0; i<staffList.length; i++){  
 if(staffList[i] == null) {  
 staffList[i] = staff;  
 break;  
 }  
 }  
 }  
  
 double payDay(){  
 double allPaid = 0;  
 for (StaffMember person:staffList) {  
 if(person!=null) {  
 System.*out*.println(person.toString());  
 System.*out*.println();  
 allPaid += person.pay();  
 }  
 else continue;  
 }  
 return allPaid;  
 }  
}

## StaffMember:

public class StaffMember {  
 protected String name;  
 protected String surname;  
 protected String phone;  
  
 StaffMember(String \_name, String \_surname, String \_phone){  
 name = \_name;  
 surname = \_surname;  
 phone = \_phone;  
 }  
  
 public String toString(){  
 String data = "";  
 data+= name + " " + surname + " " + phone + "\n";  
 data+="Sumokėta: " + pay();  
 return data;  
 }  
  
 public double pay(){  
 return 0;  
 }  
}

## Trainee:

public class Trainee extends StaffMember{  
  
 Trainee(String \_name, String \_surname, String \_phone) {  
 super(\_name, \_surname, \_phone);  
 }  
}

## Employee:

public class Employee extends StaffMember{  
 int socInsuranceNr;  
 double salary;  
  
 Employee(String \_name, String \_surname, String \_phone, int \_socInsuranceNr, double \_salary) {  
 super(\_name, \_surname, \_phone);  
 socInsuranceNr = \_socInsuranceNr;  
 salary = \_salary;  
 }  
  
 Employee(String \_name, String \_surname, String \_phone, int \_socInsuranceNr) {  
 super(\_name, \_surname, \_phone);  
 socInsuranceNr = \_socInsuranceNr;  
 }  
  
 @Override  
 public double pay() {  
 return salary;  
 }  
}

## Executive

public class Executive extends Employee{  
 double bonus = 0;  
  
 Executive(String \_name, String \_surname, String \_phone, int \_socInsuranceNr, double \_salary, double bonus) {  
 super(\_name, \_surname, \_phone, \_socInsuranceNr, \_salary);  
 awardBonus(bonus);  
 }  
  
 void awardBonus(double bonus){  
 this.bonus+=bonus;  
 }  
  
 @Override  
 public double pay() {  
 double paid = salary + bonus;  
 bonus = 0;  
 return paid;  
 }  
}

## Hourly:

public class Hourly extends Employee{  
  
 int hoursWorked;  
 double rate;  
  
 Hourly(String \_name, String \_surname, String \_phone, int \_socInsuranceNr, int \_hoursWorked, double \_rate) {  
 super(\_name, \_surname, \_phone, \_socInsuranceNr);  
 hoursWorked = \_hoursWorked;  
 rate = \_rate;  
 }  
  
 void addHours(int hours){  
 hoursWorked+=hours;  
 }  
  
 @Override  
 public double pay() {  
 double pay = rate \* hoursWorked;  
 hoursWorked = 0;  
 return pay;  
 }  
}