package InClass.Phan2;  
import java.util.Random;  
import java.util.Scanner;  
  
abstract class Robot {  
 protected int weight;  
  
 public Robot(int weight) {  
 this.weight = weight;  
 }  
  
 abstract float calculateEnergy(float distance);  
  
 abstract void showInfo();  
  
}  
  
class Pedion extends Robot {  
 private float flex;  
 public Pedion() {  
 super(20);  
 Random random = new Random();  
 int min = 1;  
 int max = 5;  
 flex = random.nextInt(max - min + 1) + min;  
 }  
  
 @Override  
 float calculateEnergy(float distance) {  
 return (weight\*distance) + (flex+1)\*distance/2;  
 }  
  
 @Override  
 void showInfo() {  
 System.*out*.println("Pedion");  
 System.*out*.println("Weight: " + weight);  
 System.*out*.println("Flex: " + flex);  
 }  
}  
  
class Zattacker extends Robot {  
 private float power;  
 public Zattacker() {  
 super(50);  
 Random random = new Random();  
 int min = 20;  
 int max = 30;  
 power = random.nextInt(max - min + 1) + min;  
 }  
 public float calculateEnergy (float distance){  
 return (weight\*distance) + power\*power\*distance;  
 }  
  
 @Override  
 void showInfo() {  
 System.*out*.println("Zattacker");  
 System.*out*.println("Weight: " + weight);  
 System.*out*.println("Power: " + power);  
 }  
}  
  
class Carrier extends Robot {  
 private float energy;  
 public Carrier() {  
 super(30);  
 Random random = new Random();  
 int min = 50;  
 int max = 100;  
 energy = random.nextInt(max - min + 1) + min;  
 }  
  
 @Override  
 float calculateEnergy(float distance) {  
 return weight\*distance + 4\*energy\*distance;  
 }  
  
 @Override  
 void showInfo() {  
 System.*out*.println("Carrier");  
 System.*out*.println("Weight: " + weight);  
 System.*out*.println("Energy: " + energy);  
 }  
}  
  
class QuanLyRobot{  
 private Carrier[] cariers;  
 private Pedion[] pedions;  
 private Zattacker[] zattackers;  
 public QuanLyRobot(){}  
 public void input(int n, int type) {  
 switch (type) {  
 case 1:  
 pedions = new Pedion[n];  
 for (int i = 0; i < n; i++) {  
 pedions[i] = new Pedion();  
 }  
 break;  
 case 2:  
 zattackers = new Zattacker[n];  
 for (int i = 0; i < n; i++){  
 zattackers[i] = new Zattacker();  
 }  
 break;  
 case 3:  
 cariers = new Carrier[n];  
 for (int i = 0; i < n; i++){  
 cariers[i] = new Carrier();  
 }  
 break;  
 }  
 }  
  
 public void displayEnergy(float distance) {  
 for (Carrier carrier : cariers) {  
 carrier.showInfo();  
 System.*out*.println("Energy of carrier: " + carrier.calculateEnergy(distance));  
 System.*out*.println("------------------------------------------------");  
 }  
 for (Pedion pedion : pedions) {  
 pedion.showInfo();  
 System.*out*.println("Energy of pedion: " + pedion.calculateEnergy(distance));  
 System.*out*.println("------------------------------------------------");  
 }  
 for (Zattacker zattacker : zattackers) {  
 zattacker.showInfo();  
 System.*out*.println("Energy of zattacker: " + zattacker.calculateEnergy(distance));  
 System.*out*.println("------------------------------------------------");  
 }  
 }  
  
 public void maxEnergy(){  
 float max = 0;  
 String type = "";  
 float cariesEnergy = 0;  
 for (Carrier carrier : cariers) {  
 cariesEnergy += carrier.calculateEnergy(10);  
 }  
 if (cariesEnergy > max){  
 max = cariesEnergy;  
 type = "Carrier";  
 }  
 float pedionEnergy = 0;  
 for (Pedion pedion : pedions) {  
 pedionEnergy += pedion.calculateEnergy(10);  
 }  
 if (pedionEnergy > max){  
 max = pedionEnergy;  
 type = "Pedion";  
 }  
 float zattackerEnergy = 0;  
 for (Zattacker zattacker : zattackers) {  
 zattackerEnergy += zattacker.calculateEnergy(10);  
 }  
 System.*out*.println("Calculate max energy: ");  
 System.*out*.println( "\nMax energy: " + max + "\nType: " + type);  
 }  
  
}  
  
public class Lab2\_3 {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 QuanLyRobot quanLyRobot = new QuanLyRobot();  
 int n;  
 System.*out*.println("Nhập số lượng robot Pedion: ");  
 n = scanner.nextInt();  
 quanLyRobot.input(n, 1);  
 System.*out*.println("Nhập số lượng robot Zattacker: ");  
 n = scanner.nextInt();  
 quanLyRobot.input(n, 2);  
 System.*out*.println("Nhập số lượng robot Carrier: ");  
 n = scanner.nextInt();  
 quanLyRobot.input(n, 3);  
 quanLyRobot.displayEnergy(10);  
 quanLyRobot.maxEnergy();  
 }  
}