Question 1:

abstract class Water{

int litre;

int bardak;

public Water(int litre, int bardak) {

this.litre = litre;

this.bardak = bardak;

}

abstract void drinkWater();

abstract void addWater();

}

class Bb extends Water{

int mineralMiktari;

public Bb(int litre, int bardak, int mineralMiktari) {

super(litre, bardak);

this.mineralMiktari = mineralMiktari;

}

@Override

void drinkWater() {

System.out.println("Mineral Suyu ictim");

this.litre = 0;

this.bardak = 0;

}

@Override

void addWater() {

System.out.println("Eve mineral suyu aldim");

}

}

public class Aa extends Water {

int phDegeri;

public Aa(int litre, int bardak, int phDegeri) {

super(litre, bardak);

this.phDegeri = phDegeri;

}

@Override

void drinkWater() {

System.out.println(phDegeri + " Ph degerli su ictim");

}

@Override

void addWater() {

System.out.println(phDegeri + " PH degerli su aldim");

}

}

class WaterTest{

public static void main(String[] args) {

Bb b = new Bb(2,4,5);

Aa a = new Aa(1,5,7);

b.drinkWater();

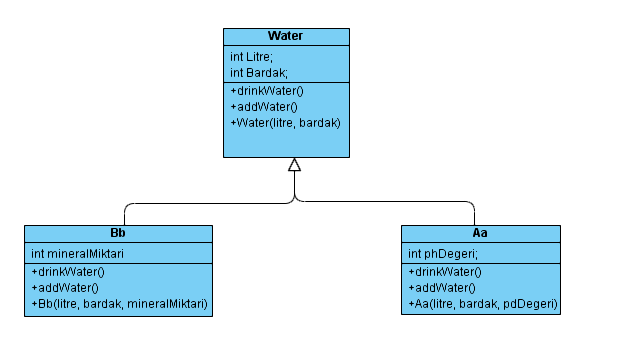
b.addWater();

a.addWater();

a.drinkWater();

}

}



interface JudoAttackBehaviours{

Fighter judoAttackTo(Fighter opponent);

}

interface MuayThaiAttackBehaviour{

Fighter muayThaiAttackTo(Fighter opponent);

}

interface BoxingAttackBehaviour{

Fighter boxingAttackTo(Fighter opponent);

}

abstract class Fighter{

String name;

int health;

public Fighter(int health, String name) {

this.name = name;

this.health = health;

}

abstract Fighter attackTo(Fighter opponent);

abstract void defense();

}

public class a {

static boolean isDead;

public static void main(String[] args) {

Fighter kibishiFighter = new KibishiFighter(100,"Ahmet");

Fighter hybridFighter = new HybridFighter(300,"Veli");

int i = 0;

isDead = false;

while(isDead != true){

if(i%2 == 0){

hybridFighter.defense();

hybridFighter = kibishiFighter.attackTo(hybridFighter);

if(hybridFighter.health <= 0){

isDead = true;

}

}

else{

kibishiFighter.defense();

kibishiFighter = hybridFighter.attackTo(kibishiFighter);

if(kibishiFighter.health <= 0){

isDead = true;

}

}

i++;

}

System.out.println("Fight is over.\n" +

"Remaining Healths are: \n" + "Kibishi Fighter: " + kibishiFighter.health + "\nHybrid Fighter: " + hybridFighter.health);

}

}

class KibishiFighter extends Fighter implements JudoAttackBehaviours{

public KibishiFighter(int health, String name) {

super(health, name);

}

@Override

Fighter attackTo(Fighter opponent) {

opponent = judoAttackTo(opponent);

return opponent;

}

@Override

public Fighter judoAttackTo(Fighter opponent) {

opponent.health -= 10;

return opponent;

}

@Override

void defense() {

this.health += 5;

}

}

class HybridFighter extends Fighter implements MuayThaiAttackBehaviour, BoxingAttackBehaviour{

public HybridFighter(int health, String name) {

super(health, name);

}

@Override

Fighter attackTo(Fighter opponent) {

int x = (int) Math.random();

if (x % 2 == 0) {

opponent = muayThaiAttackTo(opponent);

} else {

opponent = boxingAttackTo(opponent);

}

return opponent;

}

@Override

public Fighter muayThaiAttackTo(Fighter opponent) {

opponent.health -= 10;

return opponent;

}

@Override

public Fighter boxingAttackTo(Fighter opponent) {

opponent.health -= 15;

return opponent;

}

@Override

void defense() {

this.health += 5;

}

}