Jeremy Fanuel

218116757

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication8;

import java.util.ArrayList;

import java.util.Random;

import java.util.Scanner;

import java.util.StringTokenizer;

public class JavaApplication8 {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Scanner sc = new Scanner(System.in);

System.out.print("Masukkan nama : ");

String nama = sc.next();

int gold = 500;

int[] chance = {0,0,0,0,0,0,0,0,0,0,1};

ArrayList<Pokemon> poke = new ArrayList<>();

boolean gameover = false;

boolean menu1 = false;

boolean menu2 = false;

boolean menu3 = false;

boolean menu4 = false;

boolean explore = false;

boolean exploration = false;

while (!gameover)

{

System.out.println("Name: "+nama);

System.out.println("Gold: "+gold+"G");

System.out.print("Main Menu\n" + "==========================\n"+

"1. Buy Pokemon\n" +

"2. See Pokemon\n" +

"3. Evolve Pokemon\n" +

"4. Heal Pokemon\n" +

"5. Explore\n" +

"6. Cheat\n" +

"e. Exit\n"+

">>> ");

String menu = sc.next();

if(menu.equals("e"))

{

gameover = true;

}

else

{

if(menu.equals("1"))

{

menu1 = true;

while(menu1)

{

System.out.print("Buy Pokemon\n"+

"Gold : "+ gold + "\n"+

"==========================\n" +

"1. Slugma 100G\n"+

"2. Magikarp 90G\n"+

"3. Bulbasaur 80G\n"+

"b. Back\n"+

">>> ");

String pilih1 = sc.next();

if(pilih1.equals("b"))

{

menu1 = false;

}

else

{

if(pilih1.equals("1") && gold >= 100)

{

System.out.print("Insert Pokemon Name: ");

String namapoke = sc.next();

poke.add(new Slugma(namapoke));

gold -= 100;

}

else if (pilih1.equals("2") && gold >= 90)

{

System.out.print("Insert Pokemon Name: ");

String namapoke = sc.next();

poke.add(new Magikarp(namapoke));

gold -= 90;

}

else if (pilih1.equals("3") && gold >= 80)

{

System.out.print("Insert Pokemon Name: ");

String namapoke = sc.next();

poke.add(new Bulbasaur(namapoke));

gold -= 80;

}

else

{

System.out.println("Pilihan salah / uang tdk cukup");

}

}

}

}

else if (menu.equals("2"))

{

menu2 = true;

while(menu2)

{

System.out.println("See Pokemon");

System.out.println("==========================");

for (int i = 0; i < poke.size(); i++) {

System.out.println( i+1 + ". " + poke.get(i).getName());

}

System.out.println("b. Back\n>>> ");

String pilih2 = sc.next();

if(pilih2.equals("b"))

{

menu2 = false;

}

else

{

int temp\_see\_pokemon = Integer.parseInt(pilih2);

if(temp\_see\_pokemon <= poke.size())

{

int temp = Integer.parseInt(pilih2) - 1;

System.out.println(poke.get(temp).toString());

}

}

}

}

else if (menu.equals("3"))

{

menu3 = true;

while(menu3)

{

System.out.println("Evolve Pokemon");

System.out.println("==========================");

for (int i = 0; i < poke.size(); i++) {

System.out.println( i+1 +". "+poke.get(i).getName() +", type: "+poke.get(i).getType()+", level: "+poke.get(i).getLevel());

}

System.out.println("b. Back");

System.out.print(">>>");

String pilih3 = sc.next();

if(pilih3.equals("b"))

{

menu3 = false;

}

else

{

int temp\_evolve = Integer.parseInt(pilih3);

if(temp\_evolve <= poke.size())

{

int temp = Integer.parseInt(pilih3) - 1;

if(poke.get(temp).getType().equals("Magikarp"))

{

if(poke.get(temp).getLevel() < 10)

{

System.out.println("Reach level 10 first");

}

else

{

System.out.println("Evolve " +poke.get(temp).getName()+" ("+poke.get(temp).getType()+") "+"to Gyarados?(y/n)");

System.out.print(">>> ");

String confirmation = sc.next();

if(confirmation.equalsIgnoreCase("y") && poke.get(temp).getLevel() >= 10)

{

String tempname = poke.get(temp).getName();

poke.set(temp, new Gyarados(tempname));

}

}

}

else if(poke.get(temp).getType().equals("Bulbasaur"))

{

if (poke.get(temp).getLevel() < 5)

{

System.out.println("Reach level 5 first");

}

else

{

System.out.println("Evolve " +poke.get(temp).getName()+" ("+poke.get(temp).getType()+") "+"to Ivysaur?(y/n)");

System.out.print(">>> ");

String confirmation = sc.next();

if(confirmation.equalsIgnoreCase("y") && poke.get(temp).getLevel() >= 5)

{

String tempname = poke.get(temp).getName();

poke.set(temp, new Ivysaur(tempname));

}

}

}

else if(poke.get(temp).getType().equals("Ivysaur"))

{

if (poke.get(temp).getLevel() < 12)

{

System.out.println("Reach level 12 first");

}

else

{

System.out.println("Evolve " +poke.get(temp).getName()+" ("+poke.get(temp).getType()+") "+"to Venusaur?(y/n)");

System.out.print(">>> ");

String confirmation = sc.next();

if(confirmation.equalsIgnoreCase("y") && poke.get(temp).getLevel() >= 12)

{

String tempname = poke.get(temp).getName();

poke.set(temp, new Venusaur(tempname));

}

}

}

else

{

System.out.println("This pokemon can’t be evolved");

}

}

}

}

}

else if (menu.equals("4"))

{

menu4 = true;

while(menu4)

{

System.out.println("Heal Pokemon");

System.out.println("Gold : "+ gold + "G");

System.out.println("==========================");

for (int i = 0; i < poke.size(); i++) {

System.out.println(i+1+". "+poke.get(i).getName() +", HP "+poke.get(i).getHp()+"/"+poke.get(i).getMaxHP()+", MP "+ poke.get(i).getMp() + "/" + poke.get(i).getMaxMP());

}

System.out.println("a. Heal all");

System.out.println("b. Back");

System.out.print(">>> ");

String pilih4 = sc.next();

if(pilih4.equalsIgnoreCase("a"))

{

int temppokemon = 0;

for (int i = 0; i < poke.size(); i++) {

if(poke.get(i).getHp() != poke.get(i).getMaxHP())

{

temppokemon += 1;

}

}

int bayar = temppokemon \* 50;

if(gold >= bayar)

{

gold -= bayar;

for (int i = 0; i < poke.size(); i++) {

if(poke.get(i).getHp() != poke.get(i).getMaxHP())

{

poke.get(i).setHp(poke.get(i).getMaxHP());

}

}

}

else

{

System.out.println("Kekurangan gold");

}

}

else if (pilih4.equalsIgnoreCase("b"))

{

menu4 = false;

}

else

{

int temp\_heal = Integer.parseInt(pilih4);

int tempheal = Integer.parseInt(pilih4)-1;

if(temp\_heal <= poke.size())

{

if(gold >= 50 && poke.get(tempheal).getHp() != poke.get(tempheal).getMaxHP())

{

gold -= 50;

poke.get(tempheal).setHp(poke.get(tempheal).getMaxHP());

}

else if (poke.get(tempheal).getHp() == poke.get(tempheal).getMaxHP())

{

System.out.println("Health sudah max");

}

else if (gold < 50)

{

System.out.println("Gold Kurang");

}

}

}

}

}

else if (menu.equals("5"))

{

Random rand = new Random();

explore = true;

while (explore)

{

System.out.println("Pick Pokemon");

System.out.println("==========================");

for (int i = 0; i < poke.size(); i++) {

System.out.println(i+1 + ". "+poke.get(i).getName());

}

System.out.println("b. Back");

System.out.print(">>> ");

String pilihexplore = sc.next();

if(pilihexplore.equalsIgnoreCase("b"))

{

explore = false;

}

else

{

ArrayList<Integer> arrindex = new ArrayList<>();

ArrayList<String> countterarr = new ArrayList<>();

StringTokenizer tokeniser = new StringTokenizer(pilihexplore,",");

while(tokeniser.hasMoreElements() && countterarr.size() < 2)

{

countterarr.add(tokeniser.nextToken());

}

for (int i = 0; i < countterarr.size(); i++) {

arrindex.add(Integer.parseInt(countterarr.get(i)));

}

if(arrindex.size() < 2)

System.out.println("You need at least 2 pokemons");

else if(arrindex.size() >= 2)

{

exploration = true;

String[][] map = new String[12][12];

int playerx=1,playery=1;

while(exploration)

{

for (int i = 0; i < 12; i++) {

for (int j = 0; j < 12; j++) {

if(i==0||i==11||j==0||j==11)

map[i][j] = "#";

else

map[i][j] = " ";

}

}

map[playery][playerx] = "P";

for (int i = 0; i < 12; i++) {

for (int j = 0; j < 12; j++) {

System.out.print(map[i][j]);

}

System.out.println();

}

String coba = sc.next();

if(coba.equalsIgnoreCase("w"))

{

if(!map[playery-1][playerx].equals("#"))

playery--;

}

else if (coba.equalsIgnoreCase("s"))

{

if(!map[playery+1][playerx].equals("#"))

playery++;

}

else if (coba.equalsIgnoreCase("a"))

{

if(!map[playery][playerx-1].equals("#"))

playerx--;

}

else if (coba.equalsIgnoreCase("d"))

{

if(!map[playery][playerx+1].equals("#"))

playerx++;

}

else if (coba.equalsIgnoreCase("e"))

{

explore = false;

exploration = false;

}

}

}

}

}

}

else if (menu.equals("6"))

{

System.out.println("1. Gold");

System.out.println("2. Pokemon level");

int cheat = sc.nextInt();

if(cheat == 1)

{

System.out.print("Input Gold : ");

int cheatgold = sc.nextInt();

gold = cheatgold;

}

else if(cheat == 2)

{

boolean levelcheat = true;

while(levelcheat)

{

System.out.println("Cheat level");

System.out.println("===================");

for (int i = 0; i < poke.size(); i++) {

System.out.println(i+1+". "+ poke.get(i).getName()+", "+"level "+poke.get(i).getLevel());

}

System.out.println("b. Back");

String cheatmenu = sc.next();

if(cheatmenu.equalsIgnoreCase("b"))

{

levelcheat = false;

}

else

{

System.out.print("Skip level ammount: ");

int skiplevel = sc.nextInt();

int tempcheat = Integer.parseInt(cheatmenu) - 1;

poke.get(tempcheat).setLevel(poke.get(tempcheat).getLevel()+skiplevel);

}

}

}

//=========================DEBUG=====================

else if (cheat == 3)

{

for (int i = 0; i < poke.size(); i++) {

System.out.println(i+1+". "+ poke.get(i).getName() +" HP : "+poke.get(i).getHp());

}

int debughp = sc.nextInt();

poke.get(debughp - 1).setHp(poke.get(debughp - 1).getHp()-10);

}

//=========================DEBUG=====================

}

}

}

}

}

class Pokemon

{

protected String name,element,type,skill;

protected int hp,mp,maxHP,maxMP,atk,def,level;

public Pokemon(String name) {

this.name = name;

this.level = 1;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public void HurtMe()

{

this.hp -= 10;

}

public String getElement() {

return element;

}

public void setElement(String element) {

this.element = element;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getSkill() {

return skill;

}

public void setSkill(String skill) {

this.skill = skill;

}

public int getHp() {

return hp;

}

public void setHp(int hp) {

this.hp = hp;

}

public int getMp() {

return mp;

}

public void setMp(int mp) {

this.mp = mp;

}

public int getMaxHP() {

return maxHP;

}

public void setMaxHP(int maxHP) {

this.maxHP = maxHP;

}

public int getMaxMP() {

return maxMP;

}

public void setMaxMP(int maxMP) {

this.maxMP = maxMP;

}

public int getAtk() {

return atk;

}

public void setAtk(int atk) {

this.atk = atk;

}

public int getDef() {

return def;

}

public void setDef(int def) {

this.def = def;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

@Override

public String toString() {

return "\n"+this.name +

"\n"+

"Type : " + this.type +"\n" +

"Level : " + this.level + "\n"+

"HP : " + this.hp +"/"+this.maxHP+"\n"+

"MP : " + this.mp +"/"+this.maxMP+"\n"+

"Atk : " + this.atk + "\n"+

"Def : " + this.def + "\n"+

"Element : " + this.element + "\n"+

"Skill : " + this.skill+"\n";

}

}

class Slugma extends Pokemon

{

public Slugma(String name) {

super(name);

this.maxHP = 120;

this.maxMP = 100;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk = 20;

this.def = 10;

this.element = "Fire";

this.type = "Slugma";

this.skill = "Fire breath";

}

}

class Magikarp extends Pokemon

{

public Magikarp(String name) {

super(name);

this.maxHP = 110;

this.maxMP = 120;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk = 18;

this.def = 12;

this.element = "Water";

this.type = "Magikarp";

this.skill = "Water splash";

}

}

class Bulbasaur extends Pokemon

{

public Bulbasaur(String name) {

super(name);

this.maxHP = 115;

this.maxMP = 115;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk = 15;

this.def = 5;

this.element = "Grass";

this.type = "Bulbasaur";

this.skill = "";

}

}

class Gyarados extends Magikarp

{

protected String AditionalSkill;

public Gyarados(String name) {

super(name);

this.maxHP += 20;

this.maxMP += 10;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk += 10;

this.def += 10;

this.type = "Gyarados";

this.AditionalSkill = "Tsunami waves";

}

@Override

public String toString() {

return super.toString() + "Aditional Skill : " + this.AditionalSkill;

}

}

class Ivysaur extends Bulbasaur

{

protected String AditionalSkill;

public Ivysaur(String name) {

super(name);

this.maxHP += 10;

this.maxMP += 10;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk += 5;

this.def += 5;

this.type = "Ivysaur";

this.AditionalSkill = "Razor leaf";

}

@Override

public String toString() {

return super.toString() + "Aditional Skill : " + this.AditionalSkill;

}

}

class Venusaur extends Ivysaur

{

public Venusaur(String name) {

super(name);

this.maxHP += 15;

this.maxMP += 15;

this.hp = this.maxHP;

this.mp = this.maxMP;

this.atk += 10;

this.def += 10;

this.type = "Venusaur";

this.AditionalSkill = "Seed bomb";

}

@Override

public String toString() {

return super.toString();

}

}

//class Explore {

// protected int x,y;

// public Explore() {

// this.x=1;

// this.y=1;

// }

// public void initMap()

// {

//

// }

// public void updateMap()

// {

//

// }

// public void printMap()

// {

//// for (int i = 0; i < 12; i++) {

//// for (int j = 0; j < 12; j++) {

//// if(i==0||i==11||j==0||j==11)

//// map[i][j] = "#";

//// else

//// map[i][j] = " ";

//// }

//// }

//// map[playery][playerx] = "P";

//// for (int i = 0; i < 12; i++) {

//// for (int j = 0; j < 12; j++) {

//// System.out.print(map[i][j]);

//// }

//// System.out.println();

//// }

//// String coba = sc.next();

//// if(coba.equalsIgnoreCase("w"))

//// {

//// if(!map[playery-1][playerx].equals("#"))

//// playery--;

//// }

//// else if (coba.equalsIgnoreCase("s"))

//// {

//// if(!map[playery+1][playerx].equals("#"))

//// playery++;

//// }

//// else if (coba.equalsIgnoreCase("a"))

//// {

//// if(!map[playery][playerx-1].equals("#"))

//// playerx--;

//// }

//// else if (coba.equalsIgnoreCase("d"))

//// {

//// if(!map[playery][playerx+1].equals("#"))

//// playerx++;

//// }

//// else if (coba.equalsIgnoreCase("e"))

//// {

//// explore = false;

//// exploration = false;

//// }

// }

//}