#include <iostream>

#include <vector>

#include <time.h>

using namespace std;

class **Farm**;

class **Hunter**;

class **Duck**{

protected:

enum **kinds**{Peganki=1,Marble\_tears,Peganki2,Broad\_nose,Blue\_winged\_teal,Gogoli}kinds;

string name,tail\_color,wing\_shape,favourite\_dish,eye\_color;

int heft=0,old=0,height=0,wingspan=0,force=0;

bool run=0,swim=0,fight=0,lounging=0,fly=0,know=0;

public:

**Duck**(int a){

switch (a) {

case 0:{

//Утка вид: 'Пеганки', умеет: летать и знает где она живет, а также имеет атрибуты:имя, вес, окрас хвоста, форма крыльев

kinds=Peganki;

fly=1;

know=1;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2; //Вес утки

string color[2] = {"Обычный", "Необычный"};

tail\_color=color[rand() % 2];

string wings[2] = {"Обычные", "Необычные"};

wing\_shape=wings[rand() % 2];

break;

}

case 1:{ //Утка вид: 'Мраморные чирки', умеет: бегать и не знает где она живет, а также имеет атрибуты: имя, вес, возраст, окрас хвоста

kinds=Marble\_tears;

know=0;

run=1;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2;

old= rand() % 5 +2;

string color[2] = {"Обычный", "Необычный"};

tail\_color=color[rand() % 2];

break;

}

case 2:{ //Утка вид: 'Пеганки', умеет: летать и не знает где она живет, а также имеет атрибуты:имя, вес, форма крыльев, высота

kinds=Peganki2;

fly=1;

know=0;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2;

string wings[2] = {"Обычные", "Необычные"};

wing\_shape=wings[rand() % 2];

height=rand() % 10 + 50;

break;

}

case 3:{ //Утка вид: 'Широконоска', умеет: бездельничать и не знает где она живет, а также имеет атрибуты: имя, вес, форма крыльев, любимое блюдо

kinds=Broad\_nose;

know=0;

lounging=1;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2;

string wings[2] = {"Обычные", "Необычные"};

wing\_shape=wings[rand() % 2];

string dish[2] = {"Лягушки", "Насекомые"};

favourite\_dish=dish[rand() % 2];

break;

}

case 4:{ //Утка вид: 'Голубокрылый чирок', умеет: плавать и не знает где она живет, а также имеет атрибуты: имя, вес, размах крыльев, окрас крыльев

kinds=Blue\_winged\_teal;

know=0;

swim=1;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2;

wingspan=rand() % 20 + 80;

string color[2] = {"Обычный", "Необычный"};

tail\_color=color[rand() % 2];

break;

}

case 5:{ //Утка вид: 'Гоголи', умеет: клеваться (охотники не всегда могут ее поймать) и не знает где она живет, а также имеет атрибуты: имя, вес, цвет глаз, сила

kinds=Gogoli;

know=0;

fight=1;

string str[5] = {"Монти", "Шарлемань", "Диллард", "Руперт", "Джек" };

name=str[rand() % 5]; //Имя утки

heft=rand() % 2 + 2;

string color[5] = {"Синий", "Голубой", "Зелёный", "Карий", "Жёлтый" };

eye\_color=color[rand() % 5];

force=rand() % 5 +1;

break;

}

}

}

bool **GetFight**(){return fight;}

bool **GetFly**(){return fly;}

bool **GetKnow**(){return know;}

};

/\*

void Populate(){

for (int y=0;y<4;y++){

for (int z=0;z<13;z++){

Card \*b(new Card(y,z));

Add(b);

}

}

}

\*/

class **Population**{

protected:

public:

vector<Duck\*>v;

void **Add**(Duck \*duck){ //Добавить утку на озеро

v.push\_back(duck);

}

Duck\* **Get**(){ //Забрать одну утку с озера

//Проверка есть ли утки +

Duck \*n=v.back();

if (n->GetFight()){ //Проверка на дерущуюся утку

if(rand()%2==1){

v.pop\_back(); //удалить последний элемент вектора

return n;

} else return nullptr;////////////////////////////////////////////////////////////////////////////////////////////////

} else {

v.pop\_back(); //удалить последний элемент вектора

return n;

}

}

Duck\* **EscapeFree**(int n){

Duck\* m = v[n];

auto iter = v.cbegin(); // указатель на первый элемент

v.erase(iter + n);

return m;

}

bool **Check**(){ //Да, пуст

return v.empty();

}

};

class **Lake**:public Population{

public:

void **LakeIns**(int name,int a){ //заполняем озёра утками

/\* Хорниндальсватнет 1

\* Большое Невольничье озеро 2

\* Хорниндальсватнет2 3

\*/

if (name==1){

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

int c=rand()%2;

Duck \*b(new Duck(c));

Add(*b*);

}

} else if (name==2){

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

int c=rand()%2+2;

Duck \*b(new Duck(c));

Add(*b*);

}

} else {

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

int c=rand()%2+4;

Duck \*b(new Duck(c));

Add(*b*);

}

}

}

};

class **Hunter**{

protected:

int t,t1;

public:

**Hunter**(int a, int b):t(a),t1(b){};

Duck\* **Hunting**(Lake \*a){return a->Get();};

};

class **Farm** :public Population{

protected:

Hunter First,Second;

int t1,t11,t2,t21;// количество min max пойманых уток

int variant;

public:

**Farm**(int a,int b,int c,int d,int e):First(a,b),Second(c,d),t1(a),t11(b),t2(c),t21(d),variant(e){}

void **goHunting**(Lake \*a){

int n=rand() % (t11-t1)+t1;//сколько уток поймает охотник за день охоты

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

if(a->Check()){break;} //Если на озере пусто прекратить охоту

Add(*First.Hunting(a)*);

}

n=rand() % (t21-t2)+t2; //сколько уток поймает охотник за день охоты

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

if(a->Check()){break;} //Если на озере пусто прекратить охоту

Add(*Second.Hunting(a)*);

}

}

};

class **Valley**{ // Долина

private:

Farm COBYCUNE = Farm(3,5,1,7,1),CAPURI = Farm(4,7,4,5,2);//фермы с количеством уток для охотников и вариант фермы

Lake Hornindalsvatnet,Great\_Slave\_Lake,Hornindalsvatnet2;

int hsd=8; //hunting season days

int ducks=190;//количество уток

public:

void **play**(){

//Заполнить озёра утками

int ducks1,ducks2,ducks3;

ducks1=rand()%ducks;//Количество уток на Hornindalsvatnet

ducks-=ducks1;

ducks2=rand()%ducks;//Количество уток на Great\_Slave\_Lake

ducks-=ducks2;

ducks3=ducks; //Количество уток на Hornindalsvatnet2

Hornindalsvatnet.LakeIns(1,ducks1);

Great\_Slave\_Lake.LakeIns(2,ducks2);

Hornindalsvatnet2.LakeIns(3,ducks3);

//Просчитать день охоты

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

Hunt(*&COBYCUNE*);

Hunt(*&CAPURI*); //охотники сходили на охоту

Escape(&*COBYCUNE*);

Escape(&*CAPURI*); //Утки сбежали

}

}

void **Hunt**(Farm \*a){

int n;

n=rand()%3+1;

if (n==1){

a->goHunting(*&Hornindalsvatnet*);

} else if(n==2){

a->goHunting(*&Great\_Slave\_Lake*);

}else {

a->goHunting(*&Hornindalsvatnet2*);

}

}

void **Escape**(Farm \*a){

for (int i=0;i<a->v.size();i++)

{

if ((rand()%100)<5)

{

if(a->v[i]->GetFly())

{

if(a->v[i]->GetKnow()){

Duck \*y = a->EscapeFree(i);//Забираем утку;

Hornindalsvatnet.Add(*y*);

} else{

Duck \*y = a->EscapeFree(i);//Забираем утку;

int n;

n=rand()%3+1;

if (n==1){

Hornindalsvatnet.Add(*y*);

} else if(n==2){

Great\_Slave\_Lake.Add(*y*);

}else {

Hornindalsvatnet2.Add(*y*);

}

}

}

}

}

}

};

int **main**()

{

srand(time(NULL));

Valley a;

a.play();

return 0;

}