class Enemy

{

protected: // доступно наследникам

char \*name;

int hp;

int maxhp;

int regspd;

int movspd;

int dmg;

Point \*way;

public:

Enemy():hp(0), maxhp(0), regspd(0), movspd(0), dmg(0), way(NULL){};

Enemy(char \*,int, int, int, int, int);

Enemy(const Enemy &);

virtual Enemy\* stk(Land \*);

Enemy& operator=(Enemy&);

char \* getName(){return name;}

int harm(int d){return hp = hp - d;}

virtual Square \* atk(Land \*);

int getHp(){return hp;}

int getMhp(){return maxhp;}

int getRegs(){return regspd;}

int getMovs(){return movspd;}

int getDmg(){return dmg;}

int getMhp(Land \*);

int getRegs(Land \*);

int getMovs(Land \*);

int getDmg(Land \*);

char \* setName(char \*n){return name = n;}

void setFts(char \*, int, int, int, int, int);

Point \* setWay(Point \*w){return way = w;}

int setHp(int h){return hp = h;}

int setMhp(int h){return maxhp = h;}

int setRegs(int r){return regspd = r;}

int setMovs(int m){return movspd = m;}

int setDmg(int d){return dmg = d;}

~Enemy();

};

class Hero

{

protected:

lstAurs \*aur;

public:

Hero(){aur = NULL;}

Hero(lstAurs \*l){aur = l;}

Hero& operator=(Hero&);

void setAur(lstAurs \*a){aur = a;}

lstAurs \* getAur(){return aur;}

void spreadAurs(int x, int y, int r, Land \*l);

};

class Tower: public Protect

{

private:

int lvl;

twrTbl \*tbl;

public:

Tower():lvl(0), tbl(NULL){};

Tower(twrTbl\* t):lvl(0){tbl = t;}

Tower(int l, twrTbl\* t):lvl(l){tbl = t;}

void setTbl(twrTbl\* t){tbl = t;}

int lvlUp(){return ++lvl;} // повысить уровень

int setLvl(int l){return lvl = l;}

twrTbl \* getTbl()const{return tbl;}

int getLvl()const{return lvl;}

int getRds()const{return tbl->getRds(lvl);}

int getDmg()const{return tbl->getDmg(lvl);}

int getSpd()const{return tbl->getSpd(lvl);}

int getCost()const{return tbl->getCost(lvl);}

Enemy\* atk(Land \*); //атаковать ближайшего врага

};

class Wall: public Protect

{

private:

int maxhp;

int hp;

public:

Wall():maxhp(0), hp(0){}

Wall(int h):maxhp(h), hp(h){}

int setHp(int);

int setMHp(int h){return maxhp = h;}

int getHp(){return hp;}

int getMHp(){return maxhp;}

int harm(int); //нанести урон

int rpr(int); //починить

};

class Castle: public Square //замок

{

private:

char name[10];

int lvl;

int hp;

cslTbl \*tbl;

public:

Castle():lvl(0), tbl(NULL){};

Castle(cslTbl\* t){tbl = t;}

Castle(int l, cslTbl\* t):lvl(0){tbl = t;}

Castle(int h, int l, cslTbl\* t):lvl(l), hp(h){tbl = t;}

void setTbl(cslTbl\* t){tbl = t;}

int lvlUp(){return ++lvl;} // повысить уровень

int setLvl(int l){return lvl = l;}

int setHp(int h){return hp = h;}

int getHp(){return hp;}

int harm(int dmg){return hp -= dmg;}

int getLvl()const{return lvl;}

cslTbl \* getTbl()const{return tbl;}

char \* getName(){return name;}

int getYld()const{return tbl->getYld(lvl);}

int getMhp()const{return tbl->getMhp(lvl);}

int getSpd()const{return tbl->getSpd(lvl);}

int getCost()const{return tbl->getCost(lvl);}

};

class Land //ландшафт

{

private:

int sizeX;

int sizeY;

vector<Square\*> sqr;

vector<Enemy\*> enm;

vector<vector<lstAurs\*>> aurs;

int getInx(int x, int y){return sizeX \* y + x;}

public:

Land();

Land(int x, int y);

void setSz(int x, int y);

Square\* getSqr(int x, int y);

Square\* setSqr(int x, int y, Square\*);

Enemy\* getEnm(int x, int y);

Enemy\* setEnm(int x, int y, Enemy\*);

int numAurs(int x, int y);

int addAurs(int x, int y, lstAurs\*);

void clrAurs(int x, int y);

int adlMhp(int x, int y);

int adlDmg(int x, int y);

int adlRegs(int x, int y);

int adlMovs(int x, int y);

int getX(){return sizeX;}

int getY(){return sizeY;}

void norm();

int dDay(int); // сыграть один день

void print(); //печать

int save(FILE \*);

int load(FILE \*, twrTbl \*, cslTbl \*, tblEnm \*);

};

class tblEnm // таблица врагов

{

private:

dsrEnm \*turn;

dsrEnm \*fEnm;

public:

tblEnm(){turn = fEnm = NULL;}

void addEnm(Enemy \*, double);

Enemy \* getEnm(); //вернуть очередного врага

Enemy \* getEnm(double); //вернуть очередного врага, если подошло его время выхода

void reset(){turn = fEnm;}

void print();

int save(FILE \*);

int load(FILE \*, lstAurs \*);

};

class Lair: public Square //логово

{

private:

tblEnm \*tbl;

Point \*hvWay;

Point \*flWay;

Point \*lgWay;

public:

Lair():tbl(NULL), lgWay(NULL), hvWay(NULL), flWay(NULL){};

Lair(int, int, tblEnm \*, Land \*);

tblEnm \* setTbl(tblEnm \*t){return tbl = t;}

tblEnm \* getTbl(){return tbl;}

//Point \* setlgWay(Point \*w){return lgWay = w;}

void srWays(int, int, Land \*);

Point \* getlgWay(){return lgWay;}

Point \* gethvWay(){return hvWay;}

Point \* getflWay(){return flWay;}

Enemy\* rls(Land \*, int);

};