***Министерство образования Республики Беларусь***

***Учреждение образования***

***«Брестский государственный технический университет»***

***Кафедра ИИТ***

**Лабораторная работа №6**

**По дисциплине ППвИС за III семестр**

**Тема: «Пространства имен. Обработка исключений»**

**Выполнил:**

Студент группы ИИ-15 (1)

2-го курса

Волк И. А.

**Проверил:**

Михняев А.Л.

Брест 2018

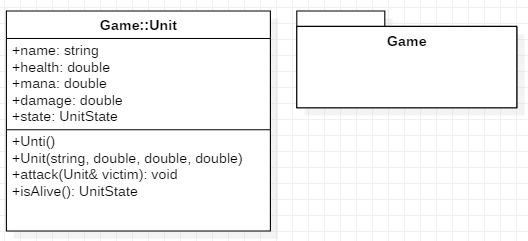
ЦЕЛЬ РАБОТЫ

Знакомство с использованием пространств имен.

Знакомство с исключениями.

Выбранная предметная область: игровой юнит.

UML:



Листинг программы:

// Application.cpp

#include <iostream>

#include "Game.h"

int main()

{

Game::Unit unit1("Bob", 100, 0, 10); // Bob

Game::Unit unit2("Tom", 100, 0, 10); // Tom

// Bob is killing Tom

while(unit2.isAlive())

unit1.attack(unit2);

Game::Unit unit3; // noname

std::cout << std::endl;

unit2.attack(unit3); // dead Bob is trying to attack noname

std::cout << std::endl;

unit1.attack(unit3); // Tom is trying to attack noname

std::cin.get();

return 0;

}

//Game.h

#ifndef \_GAME\_H\_

#define \_GAME\_H\_

#include <iostream>

#include <string>

#define UnitHealth double

#define UnitMana double

#define UnitDamage double

#define UnitName std::string

enum UnitState

{

UNIT\_DEAD = 0,

UNIT\_ALIVE

};

namespace Game

{

class Unit

{

protected:

UnitName name;

UnitHealth health;

UnitMana mana;

UnitDamage damage;

UnitState state;

public:

Unit();

Unit(UnitName n, UnitHealth h, UnitMana m, UnitDamage d);

void attack(Unit& victim) const;

UnitState isAlive() const;

};

} // namespace Game

#endif

//Game.cpp

#include "Game.h"

Game::Unit::Unit()

: name("noname"), health(0), mana(0), damage(0), state(UNIT\_DEAD) {}

Game::Unit::Unit(UnitName n, UnitHealth h, UnitMana m, UnitDamage d)

: name(n), health(h), mana(m), damage(d), state((h>0)?UNIT\_ALIVE:UNIT\_DEAD) {}

void Game::Unit::attack(Unit& victim) const

{

std::cout << name << " attack " << victim.name << "." << std::endl;

try

{

if(health == 0)

throw 0;

if(victim.health == 0)

throw 1;

if((victim.health - damage) <= 0)

throw 2;

victim.health -= damage;

std::cout << "Damage - " << damage << "." << std::endl;

}

catch(int e)

{

if(e == 0)

{

std::cout << name << " cannot attack " << victim.name << "." << std::endl;

std::cout << name << " is dead." << std::endl;

}

else if(e == 1)

{

std::cout << "Damage - 0." << std::endl;

std::cout << victim.name << " is already dead." << std::endl;

}

else if(e == 2)

{

std::cout << "Damage - " << victim.health << "." << std::endl;

std::cout << victim.name << " has been killed by " << name << "." << std::endl;

victim.health = 0;

victim.state = UNIT\_DEAD;

}

}

}

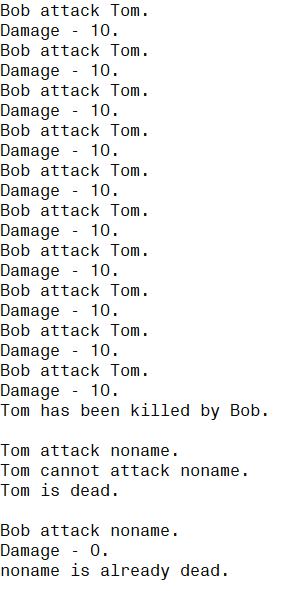
UnitState Game::Unit::isAlive() const

{

return state;

}

Результат работы:



Вывод: познакомился с использованием пространств имен и исключений.