

Домашнее задание

Выберите 3-5 видео или настольные игры. Проиграйте в каждую и проанализируйте игру, выбрав 3-5 классов и их объектов в этой игре.

Ответ на дз напишите в ворде или пдф, пример оформления ответа показан ниже в пункте «ПРИМЕР».

Задание №1: опишите полученные классы и их объекты как показано ниже в примере. Обязательно укажите объект класса с скриншотом или фото.

Класс – описывает характеристики объекта (поля – данные, методы – действия); объект – это созданный, конкретный класс.

Задание №2: напишите код для 3-5 классов и покажите их работу в программе. Пример кода есть ниже.

ПРИМЕР

Описание классов и их объектов

Я решил поиграть в видео игру Half-Life 2 и выделил такие классы и их объекты:

- 1) Класс – транспорт:
 - a. Поля: название, тип транспорта (водный, наземный, воздушный)
 - b. Методы: ехать прямо, ехать назад, повернуть влево, повернуть вправо
 - c. Объект класса: водное судно



2) Класс – оружие:

- a. Поля: название, количество боеприпасов, скорострельность, урон
- b. Методы: стрелять, перезаряжаться, убрать в инвентарь, взять из инвентаря
- c. Объект класса: пистолет



3) Класс – разрушаемый предмет

- a. Поля: название, количество здоровья, эффект разрушения
- b. Методы: получить урон, разрушится
- c. Объект класса: деревянный ящик



Код классов

```
#include <iostream>
#include <string>
#include <Windows.h>

using namespace std;

class Vehicle {
private:
    string name;
    string type;

public:
    Vehicle(string name, string type): name(name), type(type){}
    ~Vehicle() {}

    void MoveForward(){
        cout << "Транспорт - " << name << " едет вперед!" << endl;
    }

    void MoveBack() {
        cout << "Транспорт - " << name << " едет назад!" << endl;
    }

    void TurnLeft() {
        cout << "Транспорт - " << name << " повернул налево!" << endl;
    }

    void TurnRight() {
        cout << "Транспорт - " << name << " повернул направо!" << endl;
    }
};

class Weapon{
private:
    string name;
    int countAmmo;
    float fireRate;
    float damage;

public:
    Weapon(string name, int countAmmo, float fireRate, float damage):
        name(name), countAmmo(countAmmo), fireRate(fireRate), damage(damage)
    {}
    ~Weapon(){}

    float Shoot() {
        cout << "Оружие - " << name << " - стреляет с уроном: " << damage << endl;
        return damage;
    }

    void Reload() {
        cout << "Оружие - " << name << " - перезаряжается!" << endl;
    }

    void Hide() {
        cout << "Оружие - " << name << " - убрано!" << endl;
    }

    void Get() {
        cout << "Оружие - " << name << " - взято!" << endl;
    }
};
```

```

class DestroyEffect {
public:
    DestroyEffect() {}
    ~DestroyEffect(){}
    
    void PlayEffect() {
        cout << "Запуск эффекта разрушения!" << endl;
    }
};

class Damageable {
private:
    string name;
    float maxHealth;
    float currentHealth;
    DestroyEffect* destroyEffect;

public:
    Damageable(string name, float maxHealth, DestroyEffect* destroyEffect):
        name(name), maxHealth(maxHealth), currentHealth(maxHealth),
        destroyEffect(destroyEffect) {}
    ~Damageable(){}
    
    void TakeDamage(int damage) {
        currentHealth -= damage;
        
        cout << "Объект - " << name << " - получает урон: " << damage << endl
            << "Количество жизни: " << currentHealth << endl;
        
        TryDestroy();
    }

    void TryDestroy() {
        if (currentHealth <= 0)
            destroyEffect->PlayEffect();
    }
};

int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    Vehicle boat("Лодка", "Водный");
    boat.MoveForward();

    Weapon pistol("Пистолет", 12, 0.2f, 10);
    pistol.Get();

    Damageable woodBox("Деревянная коробка", 10, new DestroyEffect());
    woodBox.TakeDamage(pistol.Shoot());

    system("pause");
}

```

Пример работы

Транспорт - Лодка едет вперед!
Оружие - Пистолет - взято!
Оружие - Пистолет - стреляет с уроном: 10
Объект - Деревянная коробка - получает урон: 10
Количество жизни: 0
Запуск эфекат разрушения!
Для продолжения нажмите любую клавишу . . . ■