

C++

Урок 19

A hand-drawn blue oval frame with a double-line border, centered on the page. The text 'Разбор ДЗ' is written inside this frame in a bold, black, sans-serif font.

Разбор ДЗ

Разбор дз

1 задача:

Обычно, когда играешь в игру у персонажа можно задать только имя. Остальные поля уже заданы в момент запуска. (Конструктор по умолчанию). Мы не позволяем пользователю задавать значению полей персонажа, кроме имени.

Задача: Задайте персонажу только имя.
(Модификатор `private` у полей).

Разбор дз

1 задача:

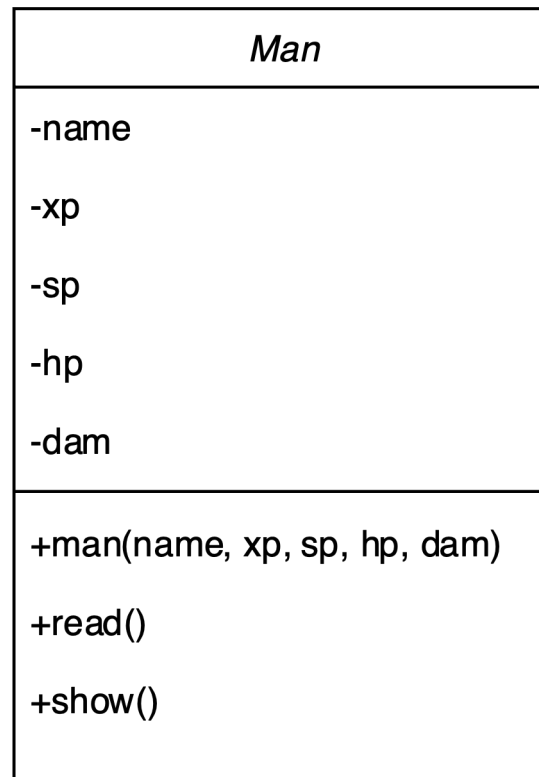


Диаграмма классов

Разбор дз

1 задача:

```
class man
{
private:
    string name;
    int xp;
    int sp;
    int hp;
    int dam;

public:
    man(string name = " ", int xp = 0, int sp = 5, int hp = 3, int dam = 20) : name(name), xp(xp), sp(sp), hp(hp), dam(dam) {}
    void set_name(string test)
    {
        name = test;
    }
    void read()
    {
        cout << "Введите имя: " << endl;
        cin >> name;
    }
    void show()
    {
        cout << "Информация об игроке: " << endl;
        cout << name << ", " << xp << " xp, " << sp << " sp, " << hp << " hp, " << dam << " dam. " << endl;
    }
};
```

Класс man

Разбор дз

1 задача:

```
int main()
{
    man A;
    A.read();
    A.show();
}
```

Main()

Разбор дз

1 задача:

```
int main()
{
    man *ptr = new man; //создали указатель типа класса на объект класса
    ptr->read();
    ptr->show();

    delete ptr; //не забываем очистить память
}
```

Main() второй вариант

Разбор дз

2 задача:

Задача: Данная задачка основывается на задаче 18.1
Реализуйте создание не одного персонажа через цикл.
Вариация решения: любой (массивы), но
предпочтительно через vector.

Разбор дз

2 задача:

```
class man
{
private:
    string name;
    int xp;
    int sp;
    int hp;
    int dam;

public:
    man(string name = " ", int xp = 0, int sp = 5, int hp = 3, int dam = 20) : name(name), xp(xp), sp(sp), hp(hp), dam(dam) {}
    void set_name(string test)
    {
        name = test;
    }
    void read()
    {
        cout << "Введите имя: " << endl;
        cin >> name;
    }
    void show()
    {
        cout << "Информация об игроке: " << endl;
        cout << name << ", " << xp << " xp, " << sp << " sp, " << hp << " hp, " << dam << " dam. " << endl;
    }
};
```

Класс man

Разбор дз

2 задача:

```
int main()
{
    int n;
    cin >> n;
    vector<man> vec(n);
    for (auto &it : vec)
    {
        it.read();
    }
    for (auto &it : vec)
    {
        it.show();
    }
}
```

Main()

Разбор дз

2 задача:

```
int main()
{
    int n;
    cin >> n;
    vector<man> vec(n);
    for (int i = 0; i < n; i++)
    {
        vec[i].show();
    }
    for (int i = 0; i < n; i++)
    {
        vec[i].show();
    }
}
```

Main() вариант 2

Разбор дз

2 задача:

```
int main()
{
    int n;
    cin >> n;
    man **ptr = new man *[n];
    //создали указатель, который указывает на массив указателей, которые создают в дин памяти объекты
    for (int i = 0; i < n; i++)
    {
        ptr[i]->read();
    }
    for (int i = 0; i < n; i++)
    {
        ptr[i]->show();
    }

    for (int i = 0; i < n; i++)
    {
        delete[] ptr[i];
    }
    delete[] ptr;
}
```

Main() вариант 3

Разбор дз

3 задача:

На данный момент должно быть следующее: можно создать персонажей одного типа, у которых уже заданы значения полей, кроме имени (его можно изменить). Давайте расширим нашего персонажа. Пусть будет: голум, фея, каменный человек, каменный человек в огне + одного придумай сам.

Задача: Нужно реализовать механизм наследования от исходного класса. Значение полей дочерних классов измените на свои. В каждом классе разные значения полей. Также нарисуй иерархию классов через диаграмму классов.

A hand-drawn blue oval frame with a slightly irregular, sketchy border, centered on the page. It contains the word 'Наследование' in bold black text.

Наследование



Что такое наследование?

Дублировали ли мы код?

наследование

Наследование — парадигма ООП, при которой дочерний объект получает те же поля и методы, что и в базовом классе.

Наследование позволяет определить базовый класс для определенных функций (доступа к данным или действий), а затем создавать производные классы, которые наследуют или переопределяют функции базового класса.

наследование

Базовый класс – тот, от которого реализуется механизм наследования

Дочерний класс – потомок базового класса.

наследование

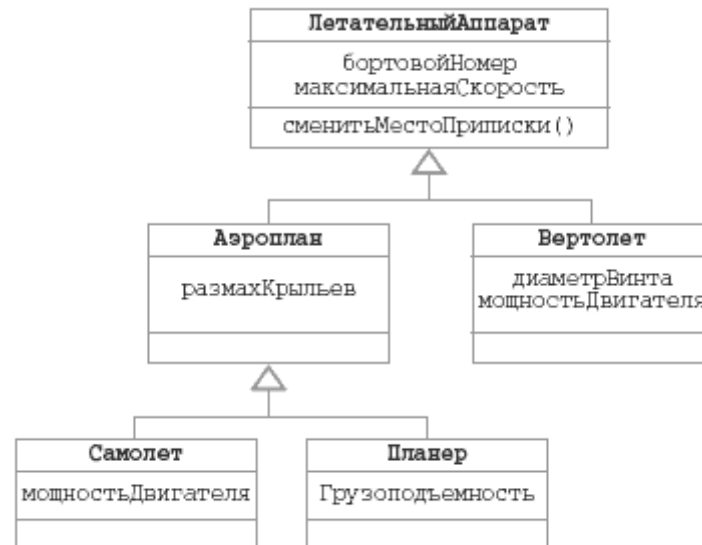
Базовый класс – тот, от которого реализуется механизм наследования

Дочерний класс – потомок базового класса. (получает те же поля и методы, что и в базовом классе)

наследование

Базовый класс – тот, от которого реализуется механизм наследования

Дочерний класс – потомок базового класса. (получает те же поля и методы, что и в базовом классе)



наследование

```
class A{  
  
};  
  
class B: [доступ] A{ //B – дочерний класс от класса A  
  
};
```

Синтаксис наследования

наследование

Модификаторы наследования:

public – публичные члены базового класса доступны. Приватные члены базового класса недоступны. Protected члены доступны внутри дочернего класса.

private – задается по умолчанию, может отсутствовать. И публичные и приватные члены базового класса недоступны.

protected – в базовом классе элементы, объявленные как protected, снаружи класса трактуются как private. Но в классах-наследниках эти поля доступны.

наследование

Модификатор доступа	Модификатор наследования		
	Private	Protected	Public
Private	Нет доступа	Нет доступа	Нет доступа
Protected	Private	Protected	Protected
Public	Private	Protected	Public