

## Домашнее задание 18 занятие

В этих задачах примените свои навыки построения структур. (классов).  
Прошу каждого: прочитать, что такое **UML диаграмма классов**. К какому заданию построить **свои диаграммы сущностей** (диаграммы классов). Софт для построения: <https://app.diagrams.net>.

По поводу модификаторов доступа (в uml отражаем слева от полей и методов): **+** - public, **~** - protected, **-** - private.

**Задача 18.1.** Обычно, когда играешь в игру у персонажа можно задать только имя. Остальные поля они уже заданы в момент нашего запуска. (Конструктор по умолчанию). Мы не позволяем пользователю задавать значению полей персонажа, кроме имени. Задайте персонажу только имя. (Модификатор private у полей).

```
Введите имя: Alex
Персонаж создан!
Его информация: Alex, 0 xp, 5 sp, 3 hp, 20 dam.
```

**Задача 18.2.** Данная задачка основывается на задаче 18.1 Реализуйте создание не одного персонажа через цикл. Вариация решения: любой (массивы), но **предпочтительно** через vector.

```
Сколько персонажей появится на карте: 3
Его информация: alex, 10 xp, 2 sp, 20 dam
Его информация: alex, 10 xp, 2 sp, 20 dam
Его информация: alex, 10 xp, 2 sp, 20 dam
```

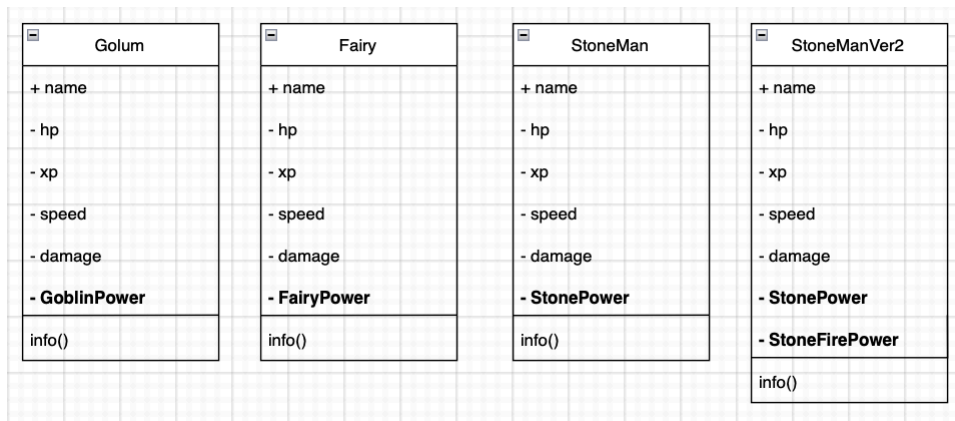
Подсказка:

```
int n;
cout << "Сколько персонажей появится на карте: ";
cin >> n;
vector <[имя класса]> my_game(n);

for (auto &it: my_game){
    it.show();
}
```

**Задача 18.3.** 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.

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.



Что main():

```
int main(){
```

```
    vector <Player> my_game(1);
    Golum obj;
    //Фея
    //Каменный
    //Каменный + огонь
```

```
    my_game.push_back(obj);
    for (auto &it: my_game){
        it.show();
    }
```

На выводе:

```
Его информация: alex, 10 xp, 2 sp, 20 dam
Его информация: Smeggle, 5 xp, 23 sp, 10 dam
```

**Задача 18.4.** Раз ты дошел до этого пункта, то молодец! Вероятно у тебя появился вопрос: **А почему мои особые поля наследуемого объекта не выводятся?** Я даже перегрузил ф-ию с новым выводом, но ничего не выходит. А почему мы вообще в вектор объектов Player записываем другие объекты? Если ты при реализации задал себе такие же вопросы - супер! Попробуй сам ответить на **выделенный вопрос**. (Подсказка: подумай про память)

Если интересно, то стоит почитать про полиморфизм и виртуальные ф-ии при наследовании, ну а если хочешь получить ответ еще раньше - то почитай про абстракцию)))