

Передача динамического типа в метод другого объекта

Рассмотрим вызов `first.Collide(second)` в реализации функции `bool Collide(const GameObject& first, const GameObject& second)`. Это виртуальный вызов. Если для каждого из четырёх имеющихся у нас классов, мы выполним его перегрузку, то внутри неё указатель `this` будет иметь тип, соответствующий динамическому типу параметра `first`. Например,

```
class Unit : public GameObject {
public:
    explicit Unit(geo2d::Point position);

    bool Collide(const GameObject& that) const override {
        // указатель this в этом методе имеет тип Unit*
    }
};
```

Следовательно, мы можем «передать» этот тип в объект `that`, вызвав метод `CollideWith`:

```
bool Unit::Collide(const GameObject& that) const {
    return that.CollideWith(*this); // Вызываем виртуальный метод GameObject::CollideWith(const
}
```

Делаем такую перегрузку для каждого класса `Unit`, `Building`, `Tower`, `Fence` и двойная диспетчеризация готова: первый виртуальный вызов мы делаем, вызывая `GameObject::Collide` из функции `Collide`, а второй — вызывая `GameObject::CollideWith(*this)` внутри переопределения метода `Collide` для **каждого** классах-наследника `GameObject`.