

## ポリモーフィズムについての補足

```
Kyara *kyara1;  
kyara1 = new Teki;
```

講義で出てきたこの 2 行ですが、何をやっているのかというと、上の 2 行のプログラムは読んで字のごとく Kyara クラス（へのポインタ）に Teki クラス（へのポインタ）を無理やり押しこんでいるだけです。

ただ、ここで 1 つ注意があります。Kyara クラスに無理やり押し込んで入るため、Kyara クラスと共通するメンバーしか使えません。つまり、Teki クラス内で新たに追加した変数、関数は使用できません。ちょっと例を見てみましょう。

< 例 >

```
class Kyara {  
public:  
    virtual void ugokasu(void) {  
    }  
};  
  
class Teki : public Kyara {  
public:  
    int taikyu; // 耐久力 (HP のようなもの) を追加  
  
    // 敵キャラ特有の攻撃を追加  
    void tokushu_kougeki() {  
        std::cout << “敵の特殊攻撃!” << std::endl;  
    }  
  
    void ugokasu() {  
        std::cout << “敵動く” << std::endl;  
    }  
};  
  
int main()  
{  
    Kyara *kyara;  
    kyara = new Teki;  
  
    kyara->taikyu = 100; // できない (Kyara クラスにそんなメンバーは存在しない)  
    kyara->tokushu_kougeki(); // できない (Kyara クラスに(略))  
    kyara->ugokasu(); // できる  
  
    return 0;  
}
```

少々ややこしいかもしれませんが、頑張って覚えましょう:(