

```
<?php
```

```
/*
```

```
 * 需求:构造不同的人,有瘦子、有胖子等不同表现的人
```

```
*/
```

```
/**
```

```
 * 人由头、身体、脚
```

```
 * 这几个部分是稳定组合算法,唯独不同的只是他们的表现,所以考虑使用"建造者模式"
```

```
 * 产品内部表现与产品生产过程分离,从而组织出不同内部表现的产品。
```

```
 * 将一个复杂对象的构造与它的表示分离,使同样的构建过程可以创建不同的表示,这样的设计模式被称为建造者模式。
```

```
*/
```

```
class person {
```

```
    private $_head;
```

```
    private $_body;
```

```
    private $_foot;
```

```
    public function setHead($head) {
```

```
        $this->_head = $head;
```

```
    }
```

```
    public function getHead() {
```

```
        return $this->_head;
```

```
    }
```

```
    public function setBody($body) {
```

```
        $this->_body = $body;
```

```
    }
```

```
    public function getBody() {
```

```
        return $this->_body;
```

```
    }
```

```
    public function setFoot($foot) {
```

```
        $this->_foot = $foot;
```

```
    }
```

```
    public function getFoot() {
```

```
        return $this->_foot;
```

```
    }
```

```

}

/**
 * 创建一个product对象的各个部件指定抽象接口
 */
abstract class PersonBuilder {
    public abstract function buildHead();
    public abstract function buildBody();
    public abstract function buildFoot();
}

class ConcreteBuilder extends PersonBuilder {
    private $_person;
    public function __construct() {
        $this->_person = new person();
    }

    public function buildHead() {
        $this->_person->setHead('建造头');
    }

    public function buildBody() {
        $this->_person->setBody('建造身体');
    }

    public function buildFoot() {
        $this->_person->setFoot('建造脚');
    }

    public function getPerson() {
        return $this->_person;
    }
}

class Diretor {
    public function construct (PersonBuilder $concreteBuilder){
        $concreteBuilder->buildHead();
        $concreteBuilder->buildBody();
        $concreteBuilder->buildFoot();
    }
}

class client {

```

```
public static function main() {  
    $builder = new ConcreteBuilder();  
    $diretor = new Diretor();  
    $diretor->construct($builder);  
    $product = $builder->getPerson();  
    var_dump($product);  
}  
}
```

```
client::main();
```