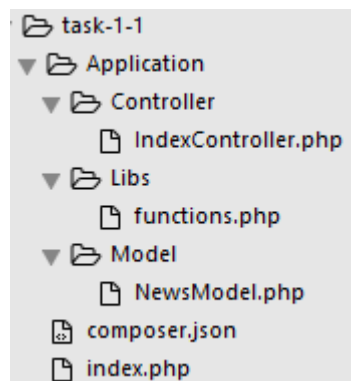


## 第 1 讲 PHP 面向对象深度剖析

### 1. 命名空间与自动加载

已知有下图所示目录结构，欲在 `/index.php` 文件中自动加载 `Application` 目录中的类（命名空间为 `App`，类文件为 `/Application/Controller/IndexController.php` 和 `/Application/Model/NewsModel.php` 等），同时自动加载 `/Application/Libs/functions.php` 文件中的辅助函数，请给出 `composer.json` 文件的核心代码。



### 2. 对象在内存中的存储

函数 `memory_get_usage()`，返回当前分配给你 PHP 脚本的内存量，单位是字节（byte）。现有以下所示代码段，请分析程序的运行结果（最好自己运行代码）。

```
class Person {
    public $name = 'test';
    public $age = 18;
}
// before
$before = memory_get_usage();
// build
$p1 = new Person();


$p2 = $p1; // 若换成 $p2 = clone $p1;，结果又如何呢？


// after
$after = memory_get_usage();
// test
echo $after - $before;
```

### 3. Traits

已知有如下代码段，请写出程序的执行结果。

```
trait TestTrait {
    public function testMethod() {
        echo "Class: " . __CLASS__ . PHP_EOL;
        echo "Trait: " . __TRAIT__ . PHP_EOL;
    }
}
```

```

}
class BaseClass {
    use TestTrait;
}
class TestClass extends BaseClass {
}

$t = new TestClass();
$t->testMethod();

```

#### 4. 后期静态绑定

已知有如下代码段，请写出程序的执行结果。若把加粗代码替换成，`return new static();`，程序的执行结果又是什么呢？

```

abstract class DomainObject {
    public static function create() {
        return new self();
    }
}
class User extends DomainObject {
}
print_r(User::create());

```