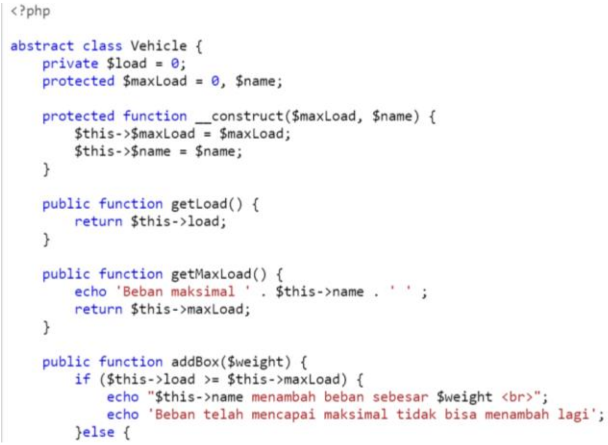
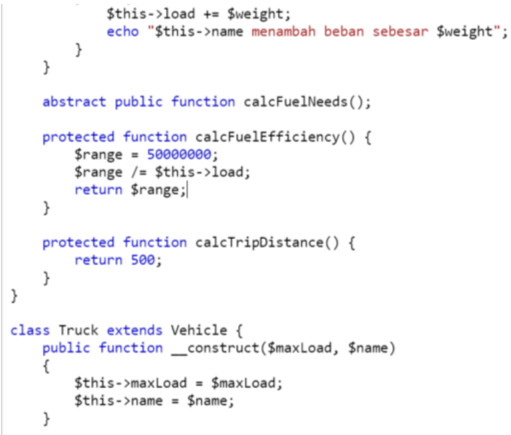
LAPORAN PRAKTIKUM 6
PEMROGRAMAN BERORIENTASI OBYEK

Cinderella Ih Hsin Chiang
Kelas B / 210091397022

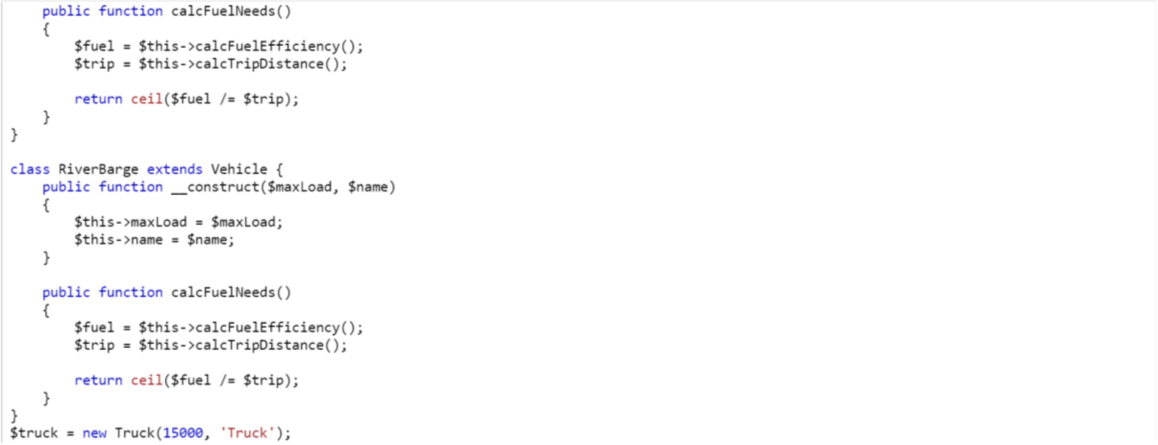
1. Source Code :



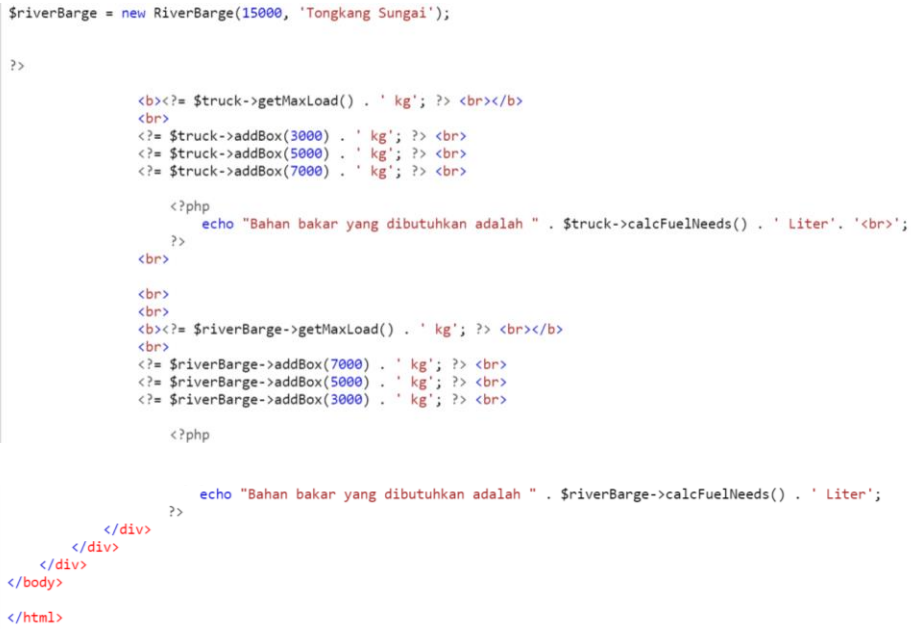
```php
<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Beban maksimal ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah beban sebesar $weight <br>";
            echo 'Beban telah mencapai maksimal tidak bisa menambah lagi';
        }else {
```



```php
            $this->load += $weight;
            echo "$this->name menambah beban sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}

class Truck extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }
```

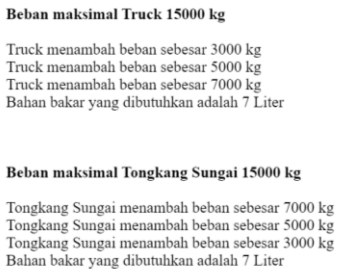


```php
    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}

class RiverBarge extends Vehicle {
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
}
$truck = new Truck(15000, 'Truck');
```



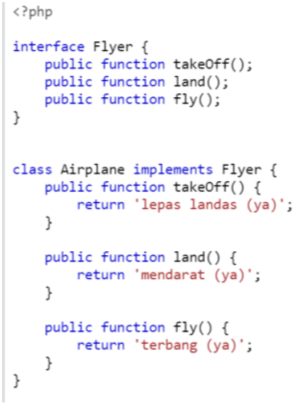
```
$riverBarge = new RiverBarge(15000, 'Tongkang Sungai');


?>

                <b><?= $truck->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $truck->addBox(3000) . ' kg'; ?> <br>
                <?= $truck->addBox(5000) . ' kg'; ?> <br>
                <?= $truck->addBox(7000) . ' kg'; ?> <br>

                    <?php
                        echo "Bahan bakar yang dibutuhkan adalah " . $truck->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
                <br>

                <br>
                <br>
                <b><?= $riverBarge->getMaxLoad() . ' kg'; ?> <br></b>
                <br>
                <?= $riverBarge->addBox(7000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(5000) . ' kg'; ?> <br>
                <?= $riverBarge->addBox(3000) . ' kg'; ?> <br>

                    <?php


                        echo "Bahan bakar yang dibutuhkan adalah " . $riverBarge->calcFuelNeeds() . ' Liter';
                    ?>
            </div>
        </div>
    </div>
</body>

</html>
```

Output :

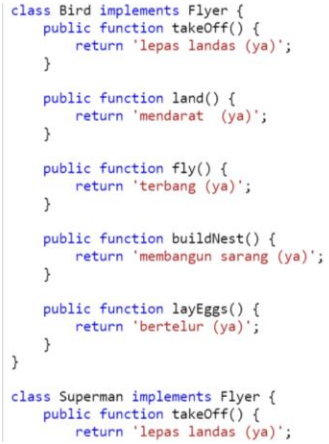


```
Beban maksimal Truck 15000 kg

Truck menambah beban sebesar 3000 kg
Truck menambah beban sebesar 5000 kg
Truck menambah beban sebesar 7000 kg
Bahan bakar yang dibutuhkan adalah 7 Liter



Beban maksimal Tongkang Sungai 15000 kg

Tongkang Sungai menambah beban sebesar 7000 kg
Tongkang Sungai menambah beban sebesar 5000 kg
Tongkang Sungai menambah beban sebesar 3000 kg
Bahan bakar yang dibutuhkan adalah 7 Liter
```

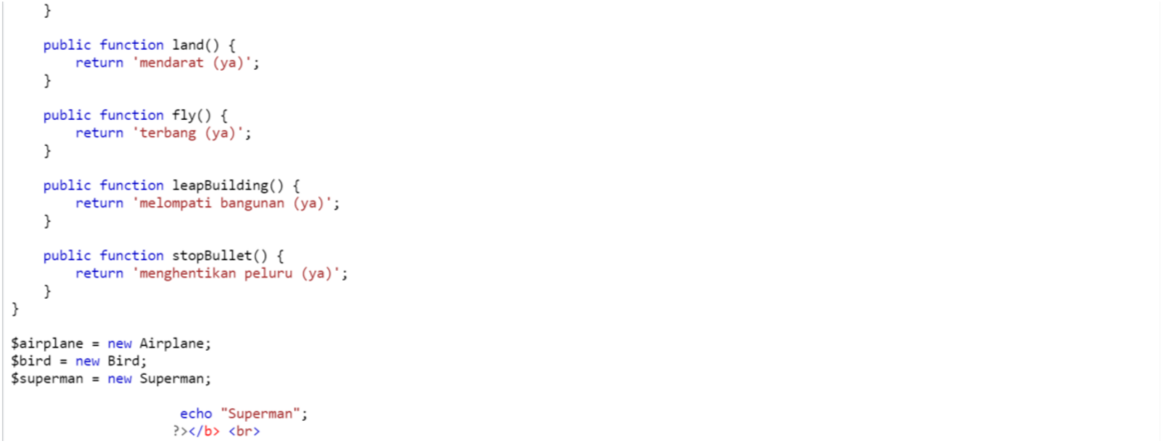
2. Source Code :



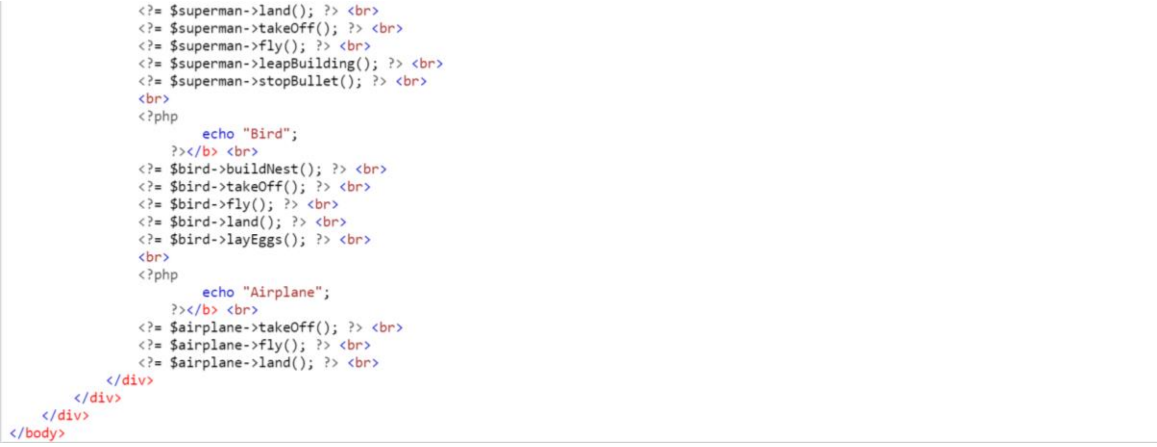
```php
<?php

interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}


class Airplane implements Flyer {
    public function takeOff() {
        return 'lepas landas (ya)';
    }

    public function land() {
        return 'mendarat (ya)';
    }

    public function fly() {
        return 'terbang (ya)';
    }
}
```



```php
class Bird implements Flyer {
    public function takeOff() {
        return 'lepas landas (ya)';
    }

    public function land() {
        return 'mendarat  (ya)';
    }

    public function fly() {
        return 'terbang (ya)';
    }

    public function buildNest() {
        return 'membangun sarang (ya)';
    }

    public function layEggs() {
        return 'bertelur (ya)';
    }
}

class Superman implements Flyer {
    public function takeOff() {
        return 'lepas landas (ya)';
```

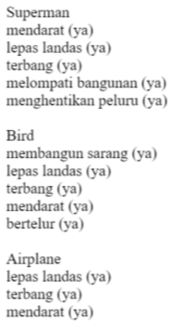


```php
    }

    public function land() {
        return 'mendarat (ya)';
    }

    public function fly() {
        return 'terbang (ya)';
    }

    public function leapBuilding() {
        return 'melompati bangunan (ya)';
    }

    public function stopBullet() {
        return 'menghentikan peluru (ya)';
    }
}

$airplane = new Airplane;
$bird = new Bird;
$superman = new Superman;

                     echo "Superman";
                    ?></b> <br>
```



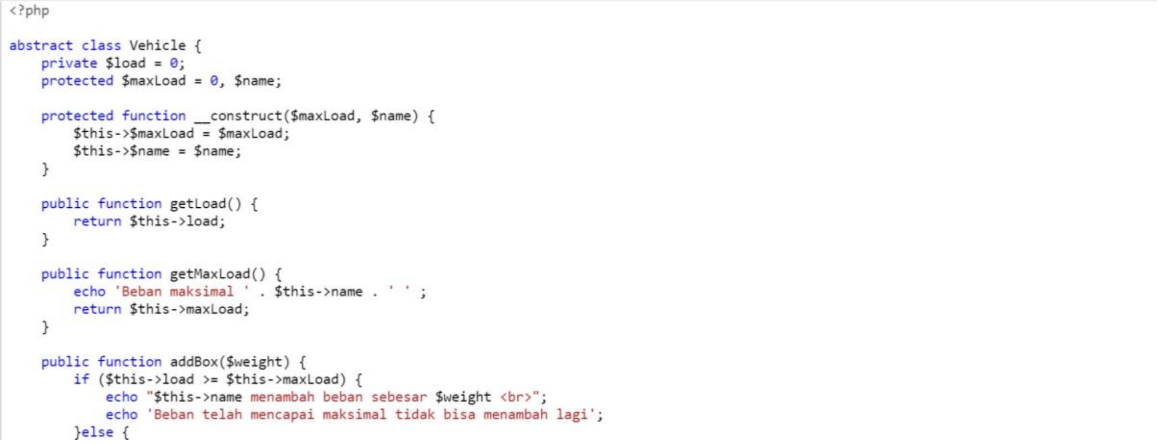
```
                <?= $superman->land(); ?> <br>
                <?= $superman->takeOff(); ?> <br>
                <?= $superman->fly(); ?> <br>
                <?= $superman->leapBuilding(); ?> <br>
                <?= $superman->stopBullet(); ?> <br>
                <br>
                <?php
                        echo "Bird";
                    ?></b> <br>
                <?= $bird->buildNest(); ?> <br>
                <?= $bird->takeOff(); ?> <br>
                <?= $bird->fly(); ?> <br>
                <?= $bird->land(); ?> <br>
                <?= $bird->layEggs(); ?> <br>
                <br>
                <?php
                        echo "Airplane";
                    ?></b> <br>
                <?= $airplane->takeOff(); ?> <br>
                <?= $airplane->fly(); ?> <br>
                <?= $airplane->land(); ?> <br>
            </div>
        </div>
    </div>
</body>
```

Output :

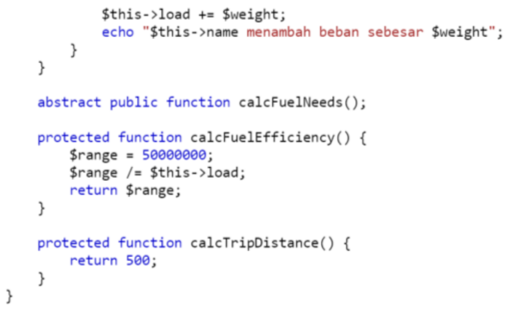


```
Superman
mendarat (ya)
lepas landas (ya)
terbang (ya)
melompati bangunan (ya)
menghentikan peluru (ya)

Bird
membangun sarang (ya)
lepas landas (ya)
terbang (ya)
mendarat (ya)
bertelur (ya)

Airplane
lepas landas (ya)
terbang (ya)
mendarat (ya)
```

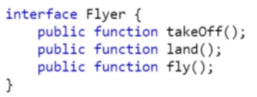
3. Source Code :



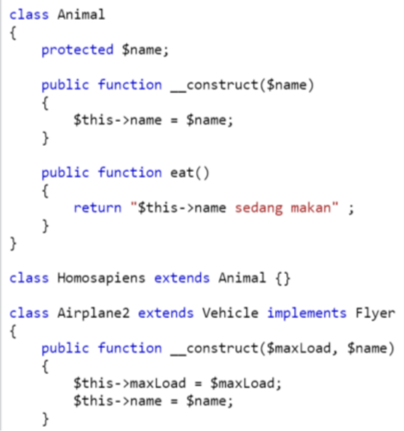
```php
<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Beban maksimal ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah beban sebesar $weight <br>";
            echo 'Beban telah mencapai maksimal tidak bisa menambah lagi';
        }else {
```



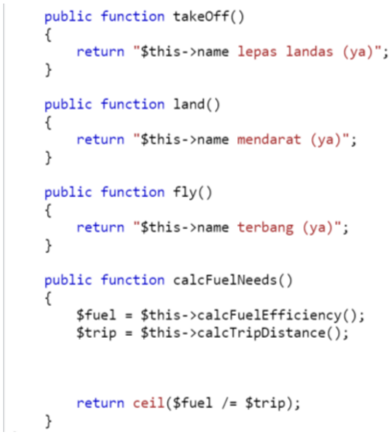
```php
            $this->load += $weight;
            echo "$this->name menambah beban sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}
```



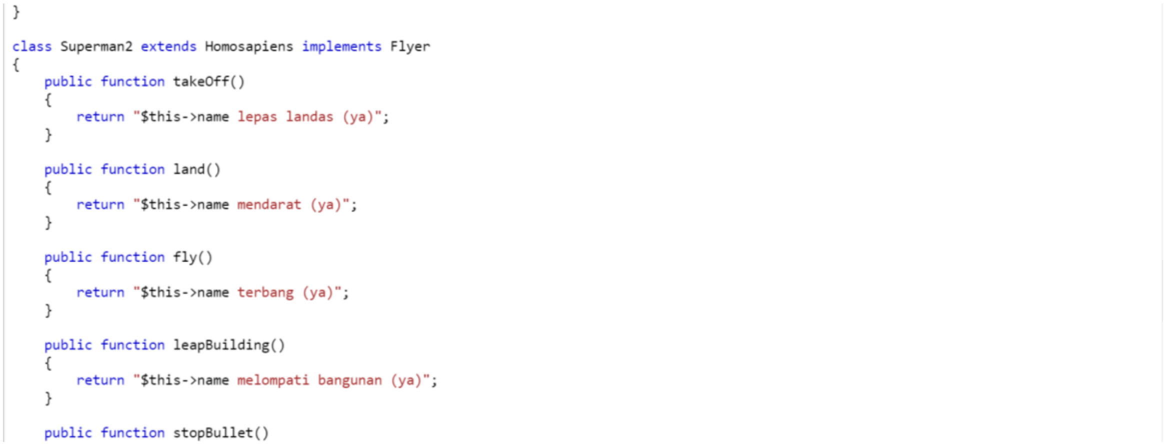
```php
interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}
```



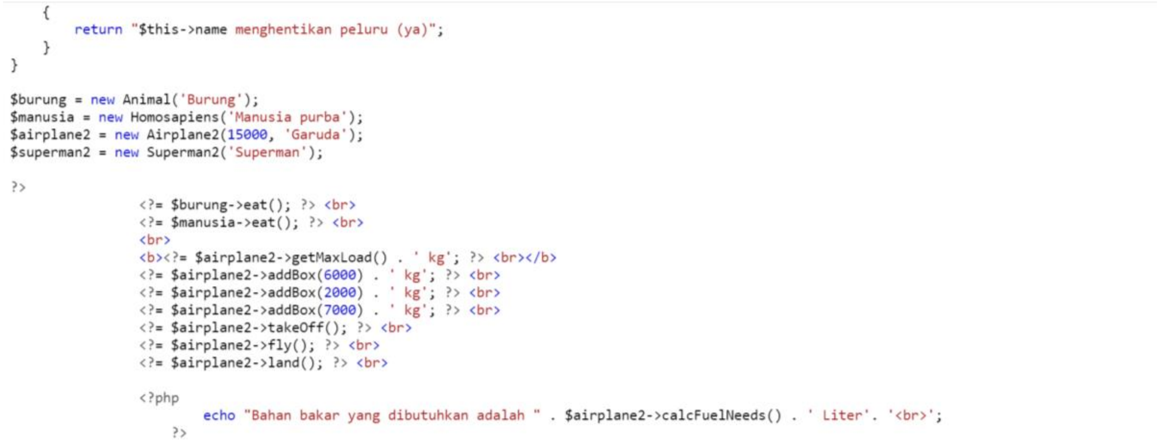
```php
class Animal
{
    protected $name;

    public function __construct($name)
    {
        $this->name = $name;
    }

    public function eat()
    {
        return "$this->name sedang makan" ;
    }
}

class Homosapiens extends Animal {}

class Airplane2 extends Vehicle implements Flyer
{
    public function __construct($maxLoad, $name)
    {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }
```



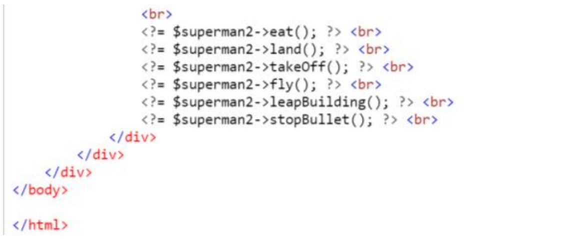
```
    public function takeOff()
    {
        return "$this->name lepas landas (ya)";
    }

    public function land()
    {
        return "$this->name mendarat (ya)";
    }

    public function fly()
    {
        return "$this->name terbang (ya)";
    }

    public function calcFuelNeeds()
    {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();



        return ceil($fuel /= $trip);
    }
```



```
}

class Superman2 extends Homosapiens implements Flyer
{
    public function takeOff()
    {
        return "$this->name lepas landas (ya)";
    }

    public function land()
    {
        return "$this->name mendarat (ya)";
    }

    public function fly()
    {
        return "$this->name terbang (ya)";
    }

    public function leapBuilding()
    {
        return "$this->name melompati bangunan (ya)";
    }

    public function stopBullet()
```

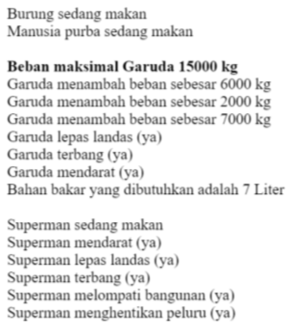


```
    {
        return "$this->name menghentikan peluru (ya)";
    }
}

$burung = new Animal('Burung');
$manusia = new Homosapiens('Manusia purba');
$airplane2 = new Airplane2(15000, 'Garuda');
$superman2 = new Superman2('Superman');

?>
                <?= $burung->eat(); ?> <br>
                <?= $manusia->eat(); ?> <br>
                <br>
                <b><?= $airplane2->getMaxLoad() . ' kg'; ?> <br></b>
                <?= $airplane2->addBox(6000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(2000) . ' kg'; ?> <br>
                <?= $airplane2->addBox(7000) . ' kg'; ?> <br>
                <?= $airplane2->takeOff(); ?> <br>
                <?= $airplane2->fly(); ?> <br>
                <?= $airplane2->land(); ?> <br>

                <?php
                        echo "Bahan bakar yang dibutuhkan adalah " . $airplane2->calcFuelNeeds() . ' Liter'. '<br>';
                    ?>
```



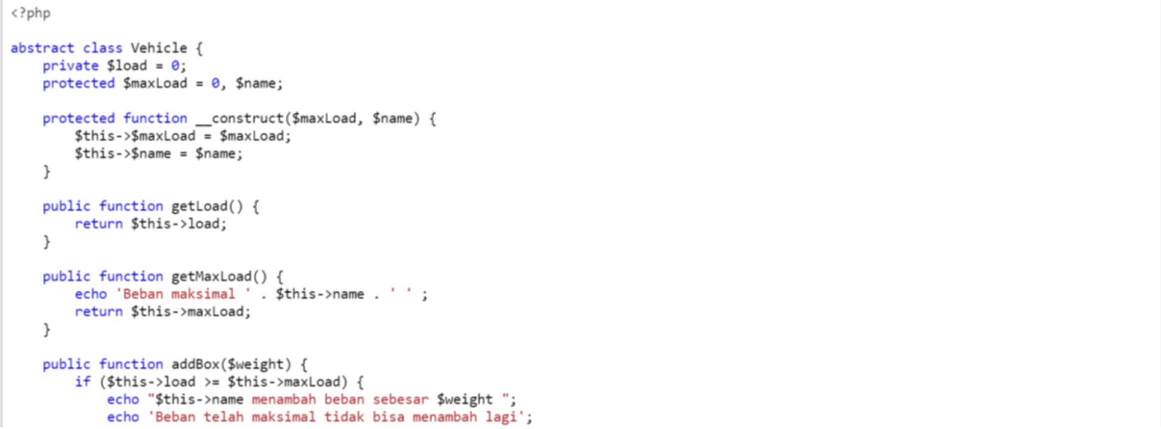
```
                <br>
                <?= $superman2->eat(); ?> <br>
                <?= $superman2->land(); ?> <br>
                <?= $superman2->takeOff(); ?> <br>
                <?= $superman2->fly(); ?> <br>
                <?= $superman2->leapBuilding(); ?> <br>
                <?= $superman2->stopBullet(); ?> <br>
            </div>
        </div>
    </div>
</body>

</html>
```

Output :

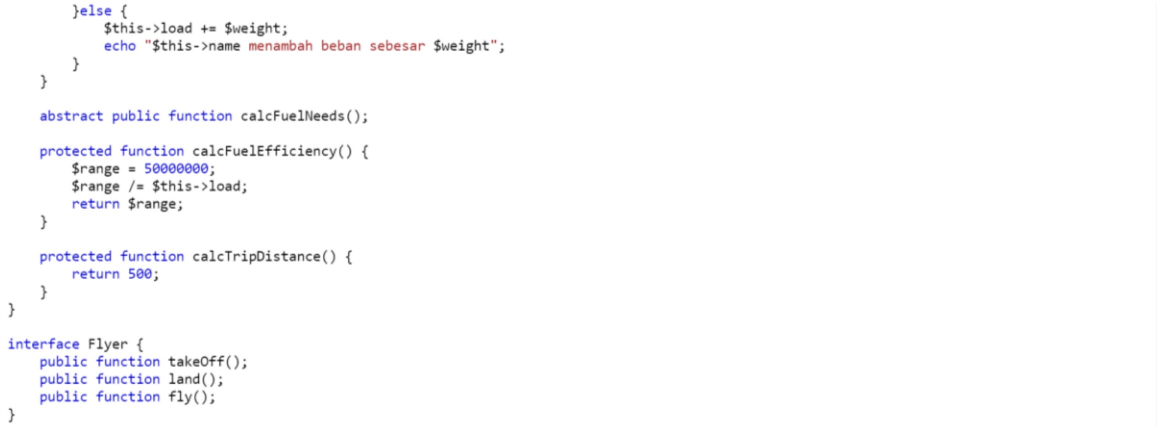


```
Burung sedang makan
Manusia purba sedang makan

Beban maksimal Garuda 15000 kg
Garuda menambah beban sebesar 6000 kg
Garuda menambah beban sebesar 2000 kg
Garuda menambah beban sebesar 7000 kg
Garuda lepas landas (ya)
Garuda terbang (ya)
Garuda mendarat (ya)
Bahan bakar yang dibutuhkan adalah 7 Liter

Superman sedang makan
Superman mendarat (ya)
Superman lepas landas (ya)
Superman terbang (ya)
Superman melompati bangunan (ya)
Superman menghentikan peluru (ya)
```

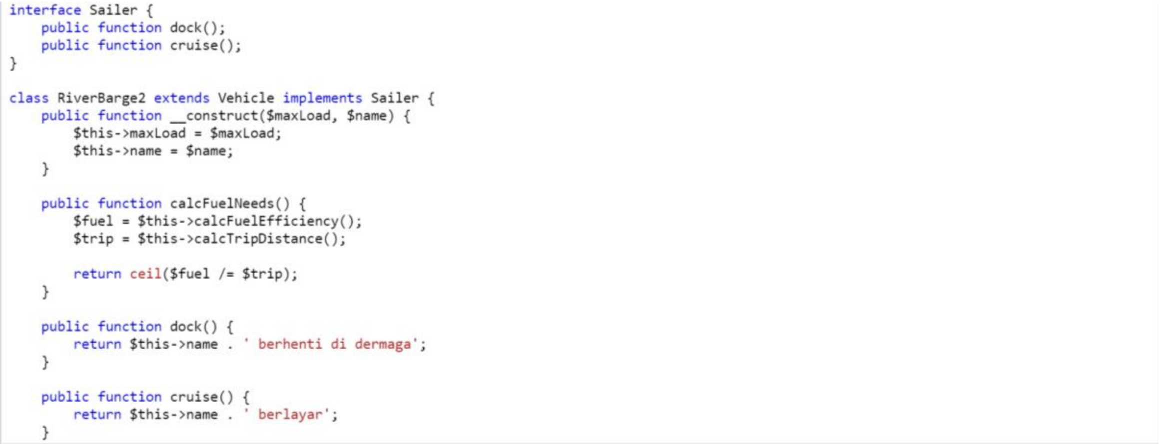
4. Source Code :



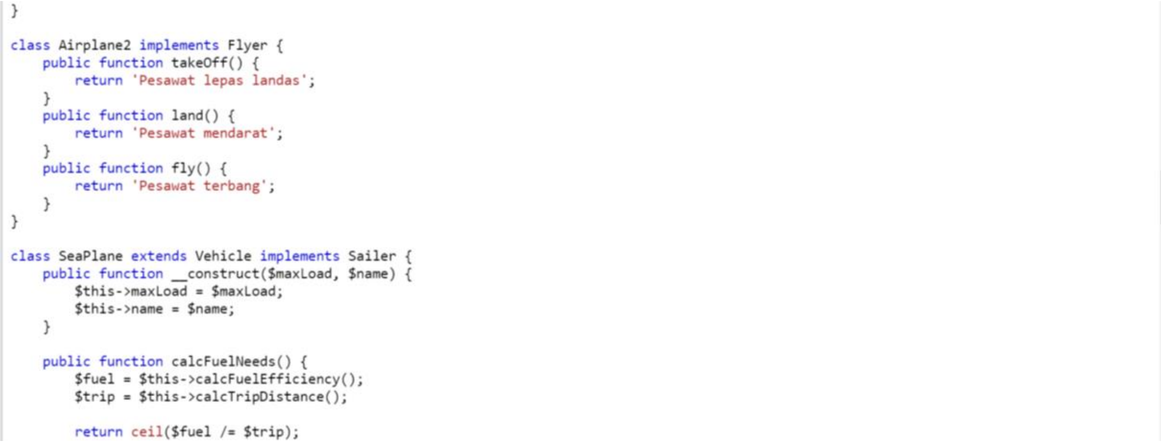
```
<?php

abstract class Vehicle {
    private $load = 0;
    protected $maxLoad = 0, $name;

    protected function __construct($maxLoad, $name) {
        $this->$maxLoad = $maxLoad;
        $this->$name = $name;
    }

    public function getLoad() {
        return $this->load;
    }

    public function getMaxLoad() {
        echo 'Beban maksimal ' . $this->name . ' ' ;
        return $this->maxLoad;
    }

    public function addBox($weight) {
        if ($this->load >= $this->maxLoad) {
            echo "$this->name menambah beban sebesar $weight ";
            echo 'Beban telah maksimal tidak bisa menambah lagi';
```



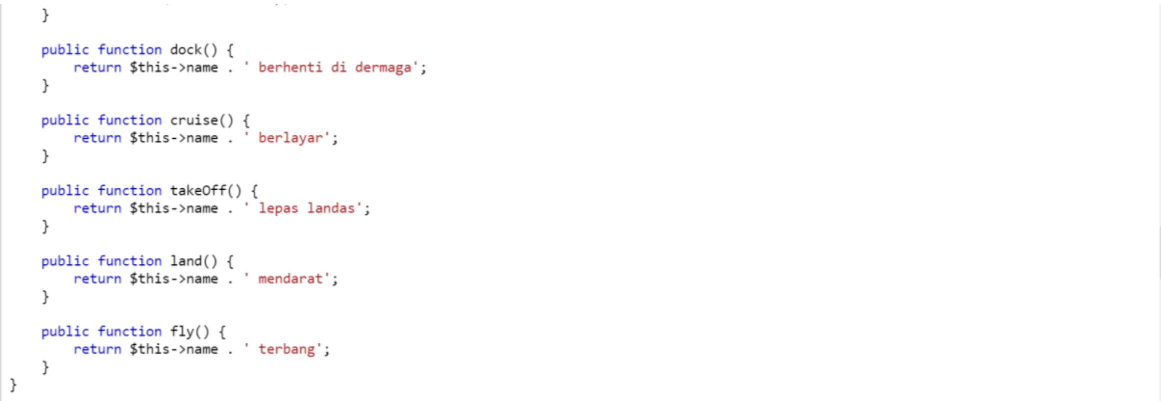
```
        }else {
            $this->load += $weight;
            echo "$this->name menambah beban sebesar $weight";
        }
    }

    abstract public function calcFuelNeeds();

    protected function calcFuelEfficiency() {
        $range = 50000000;
        $range /= $this->load;
        return $range;
    }

    protected function calcTripDistance() {
        return 500;
    }
}

interface Flyer {
    public function takeOff();
    public function land();
    public function fly();
}
```



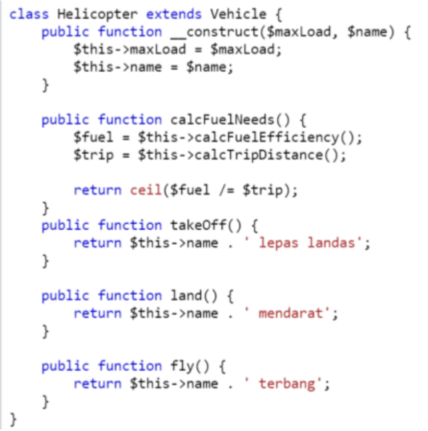
```
interface Sailer {
    public function dock();
    public function cruise();
}

class RiverBarge2 extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }

    public function dock() {
        return $this->name . ' berhenti di dermaga';
    }

    public function cruise() {
        return $this->name . ' berlayar';
    }
```



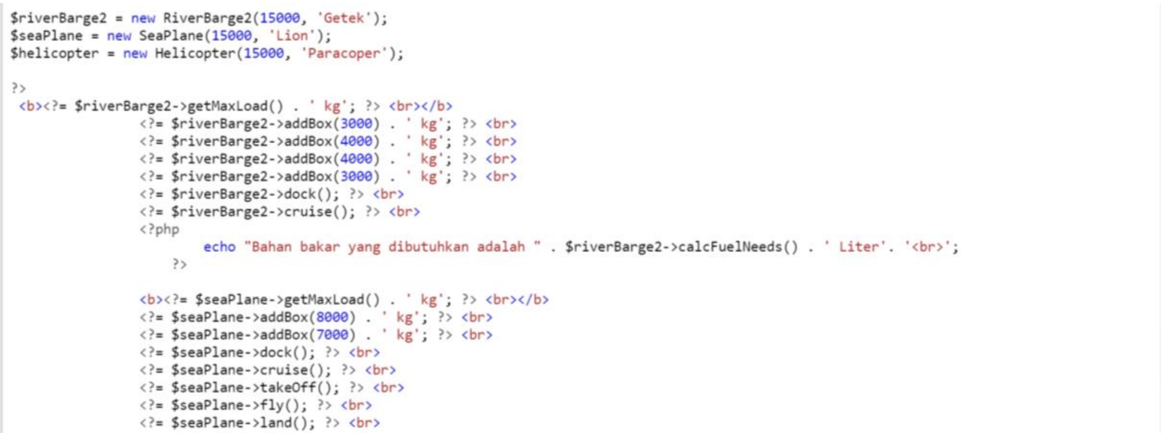
```php
}

class Airplane2 implements Flyer {
    public function takeOff() {
        return 'Pesawat lepas landas';
    }
    public function land() {
        return 'Pesawat mendarat';
    }
    public function fly() {
        return 'Pesawat terbang';
    }
}

class SeaPlane extends Vehicle implements Sailer {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
```



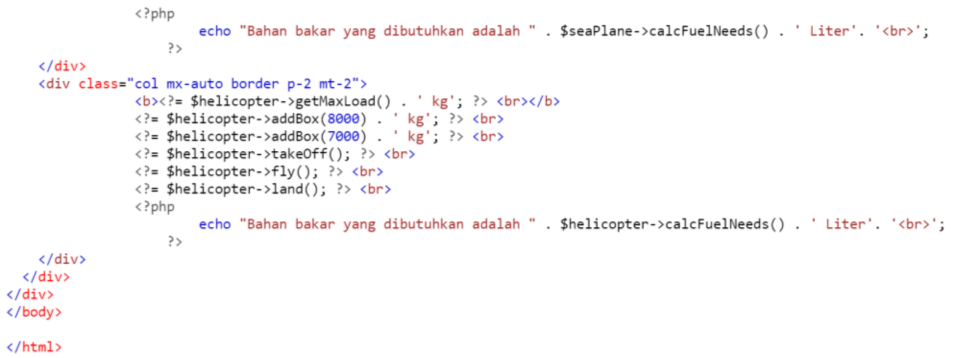
```php
    }

    public function dock() {
        return $this->name . ' berhenti di dermaga';
    }

    public function cruise() {
        return $this->name . ' berlayar';
    }

    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' terbang';
    }
}
```



```php
class Helicopter extends Vehicle {
    public function __construct($maxLoad, $name) {
        $this->maxLoad = $maxLoad;
        $this->name = $name;
    }

    public function calcFuelNeeds() {
        $fuel = $this->calcFuelEfficiency();
        $trip = $this->calcTripDistance();

        return ceil($fuel /= $trip);
    }
    public function takeOff() {
        return $this->name . ' lepas landas';
    }

    public function land() {
        return $this->name . ' mendarat';
    }

    public function fly() {
        return $this->name . ' terbang';
    }
}
```

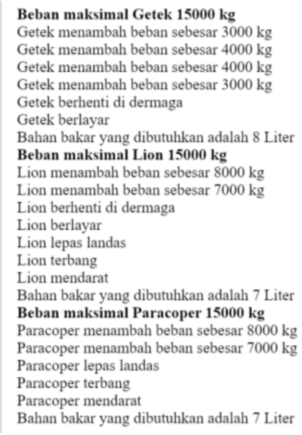


```
$riverBarge2 = new RiverBarge2(15000, 'Getek');
$seaPlane = new SeaPlane(15000, 'Lion');
$helicopter = new Helicopter(15000, 'Paracoper');

?>
 <b><?= $riverBarge2->getMaxLoad() . ' kg'; ?> <br></b>
               <?= $riverBarge2->addBox(3000) . ' kg'; ?> <br>
               <?= $riverBarge2->addBox(4000) . ' kg'; ?> <br>
               <?= $riverBarge2->addBox(4000) . ' kg'; ?> <br>
               <?= $riverBarge2->addBox(3000) . ' kg'; ?> <br>
               <?= $riverBarge2->dock(); ?> <br>
               <?= $riverBarge2->cruise(); ?> <br>
               <?php
                       echo "Bahan bakar yang dibutuhkan adalah " . $riverBarge2->calcFuelNeeds() . ' Liter'. '<br>';
                   ?>

               <b><?= $seaPlane->getMaxLoad() . ' kg'; ?> <br></b>
               <?= $seaPlane->addBox(8000) . ' kg'; ?> <br>
               <?= $seaPlane->addBox(7000) . ' kg'; ?> <br>
               <?= $seaPlane->dock(); ?> <br>
               <?= $seaPlane->cruise(); ?> <br>
               <?= $seaPlane->takeOff(); ?> <br>
               <?= $seaPlane->fly(); ?> <br>
               <?= $seaPlane->land(); ?> <br>
```



```
               <?php
                       echo "Bahan bakar yang dibutuhkan adalah " . $seaPlane->calcFuelNeeds() . ' Liter'. '<br>';
                   ?>
    </div>
    <div class="col mx-auto border p-2 mt-2">
               <b><?= $helicopter->getMaxLoad() . ' kg'; ?> <br></b>
               <?= $helicopter->addBox(8000) . ' kg'; ?> <br>
               <?= $helicopter->addBox(7000) . ' kg'; ?> <br>
               <?= $helicopter->takeOff(); ?> <br>
               <?= $helicopter->fly(); ?> <br>
               <?= $helicopter->land(); ?> <br>
               <?php
                       echo "Bahan bakar yang dibutuhkan adalah " . $helicopter->calcFuelNeeds() . ' Liter'. '<br>';
                   ?>
    </div>
  </div>
</div>
</body>

</html>
```

Output :



```
Beban maksimal Getek 15000 kg
Getek menambah beban sebesar 3000 kg
Getek menambah beban sebesar 4000 kg
Getek menambah beban sebesar 4000 kg
Getek menambah beban sebesar 3000 kg
Getek berhenti di dermaga
Getek berlayar
Bahan bakar yang dibutuhkan adalah 8 Liter
Beban maksimal Lion 15000 kg
Lion menambah beban sebesar 8000 kg
Lion menambah beban sebesar 7000 kg
Lion berhenti di dermaga
Lion berlayar
Lion lepas landas
Lion terbang
Lion mendarat
Bahan bakar yang dibutuhkan adalah 7 Liter
Beban maksimal Paracoper 15000 kg
Paracoper menambah beban sebesar 8000 kg
Paracoper menambah beban sebesar 7000 kg
Paracoper lepas landas
Paracoper terbang
Paracoper mendarat
Bahan bakar yang dibutuhkan adalah 7 Liter
```