Zad1

<?php

class NoweAuto {

protected $model;

protected $cenaEuro;

protected $kursEuroPLN;

public function \_\_construct($model, $cenaEuro, $kursEuroPLN) {

$this->model = $model;

$this->cenaEuro = $cenaEuro;

$this->kursEuroPLN = $kursEuroPLN;

}

public function ObliczCene() {

return $this->cenaEuro \* $this->kursEuroPLN;

}

}

class AutoZDodatkami extends NoweAuto {

protected $alarm;

protected $radio;

protected $klimatyzacja;

public function \_\_construct($model, $cenaEuro, $kursEuroPLN, $alarm, $radio, $klimatyzacja) {

parent::\_\_construct($model, $cenaEuro, $kursEuroPLN);

$this->alarm = $alarm;

$this->radio = $radio;

$this->klimatyzacja = $klimatyzacja;

}

public function ObliczCene() {

$cena = parent::ObliczCene();

$cena += $this->alarm + $this->radio + $this->klimatyzacja;

return $cena;

}

}

class Ubezpieczenie extends AutoZDodatkami {

protected $procentowaWartoscUbezpieczenia;

protected $liczbaLatPosiadania;

public function \_\_construct($model, $cenaEuro, $kursEuroPLN, $alarm, $radio, $klimatyzacja, $procentowaWartoscUbezpieczenia, $liczbaLatPosiadania) {

parent::\_\_construct($model, $cenaEuro, $kursEuroPLN, $alarm, $radio, $klimatyzacja);

$this->procentowaWartoscUbezpieczenia = $procentowaWartoscUbezpieczenia;

$this->liczbaLatPosiadania = $liczbaLatPosiadania;

}

public function ObliczCene() {

$cenaZDodatkami = parent::ObliczCene();

$wartoscUbezpieczenia = $this->procentowaWartoscUbezpieczenia \* ($cenaZDodatkami \* ((100 - $this->liczbaLatPosiadania) / 100));

return $wartoscUbezpieczenia;

}

}

$noweAuto = new NoweAuto("fiat 500", 30000, 4.5);

echo "Cena nowego auta w PLN: " . $noweAuto->ObliczCene() . " PLN\n";

$autoZDodatkami = new AutoZDodatkami("opel astra", 50000, 4.3, 1000, 500, 1500);

echo "Cena auta z dodatkami w PLN: " . $autoZDodatkami->ObliczCene() . " PLN\n";

$ubezpieczenie = new Ubezpieczenie("mazda 6", 35000, 2.7, 800, 400, 1200, 0.05, 2);

echo "Wartoac ubezpieczenia w PLN: " . $ubezpieczenie->ObliczCene() . " PLN\n";

?>

Zad2

<?php

class Product {

private $name;

private $price;

private $quantity;

public function \_\_construct($name, $price, $quantity) {

$this->name = $name;

$this->price = $price;

$this->quantity = $quantity;

}

public function getName() {

return $this->name;

}

public function setName($name) {

$this->name = $name;

}

public function getPrice() {

return $this->price;

}

public function setPrice($price) {

$this->price = $price;

}

public function getQuantity() {

return $this->quantity;

}

public function setQuantity($quantity) {

$this->quantity = $quantity;

}

public function \_\_toString() {

return "Product: {$this->name}, Price: {$this->price}, Quantity: {$this->quantity}";

}

}

class Cart {

private $products;

public function \_\_construct() {

$this->products = [];

}

public function addProduct(Product $product) {

$this->products[] = $product;

}

public function removeProduct(Product $product) {

foreach ($this->products as $key => $p) {

if ($p->getName() === $product->getName()) {

unset($this->products[$key]);

$this->products = array\_values($this->products);

break;

}

}

}

public function getTotal() {

$total = 0;

foreach ($this->products as $product) {

$total += $product->getPrice() \* $product->getQuantity();

}

return $total;

}

public function \_\_toString() {

$output = "Products in cart:\n";

foreach ($this->products as $product) {

$output .= $product . "\n";

}

$output .= "Total price: " . $this->getTotal() . "\n";

return $output;

}

}

$product1 = new Product("Laptop", 1500, 1);

$product2 = new Product("ekran", 500, 2);

$cart = new Cart();

$cart->addProduct($product1);

$cart->addProduct($product2);

echo $cart;

$cart->removeProduct($product1);

echo "\npo usunieciu:\n";

echo $cart;

?>