**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №2**

з дисципліни:« Проектування веб-орієнтованих інформаційних систем**»**

Виконав: студент групи ЦТ.м-51н

Кириченко Д.С.

Варіант: 8

Перевірила: Парфененко Ю.В.

Суми 2025

ТЕМА: Реалізація спадкування класів мовою PHP

Варіант 8

Завдання 2. Створити клас – кімната, що має площу. Визначити конструктор і  метод  доступу.  Створити  клас  –  однокімнатна  квартира,  що містить кімнату і кухню (їх площа), поверх.

Програмний код ФАЙЛ index2.php:

<?php  
  
// The parent class that represents a room  
class Room  
{  
 // Properties  
 protected float $area;  
 // Static property for total area in all apartments  
 private static float *$total\_area* = 0;  
  
 // Constructor  
 public function \_\_construct(float $area = 0, float $kitchen\_area = 0) {  
 $this->area = $area;  
 self::*$total\_area* += $area;  
 self::*$total\_area* += $kitchen\_area;  
 }  
  
 // Getter that uses properties like 'area' to check its existence and return value or null  
 public function \_\_get(string $property): mixed {  
 if (property\_exists($this, $property)) {  
 return $this->$property;  
 }  
 return null;  
 }  
  
 // Static function to get total area of all apartments  
 public static function getTotalArea(): float {  
 return self::*$total\_area*;  
 }  
}  
  
// The child class that represents a single one-room apartment  
class oneRoomApartment extends room  
{  
 // Properties  
 protected float $kitchen\_area;  
 protected int $room\_number;  
 protected int $floor;  
  
 //Constructor  
 public function \_\_construct(float $area = 0, float $kitchen\_area = 0, int $room\_number = 1, int $floor = 0) {  
 parent::*\_\_construct*($area, $kitchen\_area);  
 $this->kitchen\_area = $kitchen\_area;  
 $this->room\_number = $room\_number;  
 $this->floor = $floor;  
 }  
  
 public function \_\_get(string $property): mixed {  
 if (property\_exists($this, $property)) {  
 return $this->$property;  
 }  
 return null;  
 }  
}  
  
// Creating 3 objects of one-room apartments  
$one\_room1 = new OneRoomApartment(20.5, 10.4, 1, 3);  
$one\_room2 = new OneRoomApartment(12.5, 11.2, 1, 2);  
$one\_room3 = new OneRoomApartment(16.5, 10.6, 1, 5);  
  
// Calling static method to get total area  
$total\_room\_area = Room::*getTotalArea*();  
  
echo "Total room area is " . $total\_room\_area;

Завдання 4. Змініть код таким чином, щоб одержати доступ до методу test() класу B.

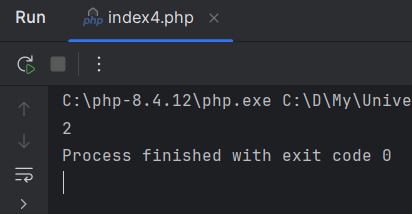


Рисунок 1 – Результат роботи програми після зміни коду

Програмний код ФАЙЛ index4.php:

<?php  
class A {  
 public static function test() {  
 echo 1;  
 }  
  
 public static function get() {  
 self::*test*();  
 }  
}  
  
class B extends A {  
 public static function test() {  
 echo 2;  
 }  
}  
  
// Now we are calling function test instead of function get that calls function test of parent class  
B::*test*();

Завдання 5. Додайте в реалізацію ієрархії класів, яку Ви розробили в завданні №2 статичне поле та статичний метод та продемонструйте роботу з ними.

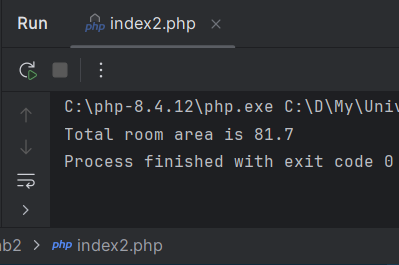


Рисунок 2 – Виведення значення статичного поля загальна площа через статичний метод

Програмний код ФАЙЛ index2.php:

<?php  
  
// The parent class that represents a room  
class Room  
{  
 // Properties  
 protected float $area;  
 // Static property for total area in all apartments  
 private static float *$total\_area* = 0;  
  
 // Constructor  
 public function \_\_construct(float $area = 0, float $kitchen\_area = 0) {  
 $this->area = $area;  
 self::*$total\_area* += $area;  
 self::*$total\_area* += $kitchen\_area;  
 }  
  
 // Getter that uses properties like 'area' to check its existence and return value or null  
 public function \_\_get(string $property): mixed {  
 if (property\_exists($this, $property)) {  
 return $this->$property;  
 }  
 return null;  
 }  
  
 // Static function to get total area of all apartments  
 public static function getTotalArea(): float {  
 return self::*$total\_area*;  
 }  
}  
  
// The child class that represents a single one-room apartment  
class oneRoomApartment extends room  
{  
 // Properties  
 protected float $kitchen\_area;  
 protected int $room\_number;  
 protected int $floor;  
  
 //Constructor  
 public function \_\_construct(float $area = 0, float $kitchen\_area = 0, int $room\_number = 1, int $floor = 0) {  
 parent::*\_\_construct*($area, $kitchen\_area);  
 $this->kitchen\_area = $kitchen\_area;  
 $this->room\_number = $room\_number;  
 $this->floor = $floor;  
 }  
  
 public function \_\_get(string $property): mixed {  
 if (property\_exists($this, $property)) {  
 return $this->$property;  
 }  
 return null;  
 }  
}  
  
// Creating 3 objects of one-room apartments  
$one\_room1 = new OneRoomApartment(20.5, 10.4, 1, 3);  
$one\_room2 = new OneRoomApartment(12.5, 11.2, 1, 2);  
$one\_room3 = new OneRoomApartment(16.5, 10.6, 1, 5);  
  
// Calling static method to get total area  
$total\_room\_area = Room::*getTotalArea*();  
  
echo "Total room area is " . $total\_room\_area;