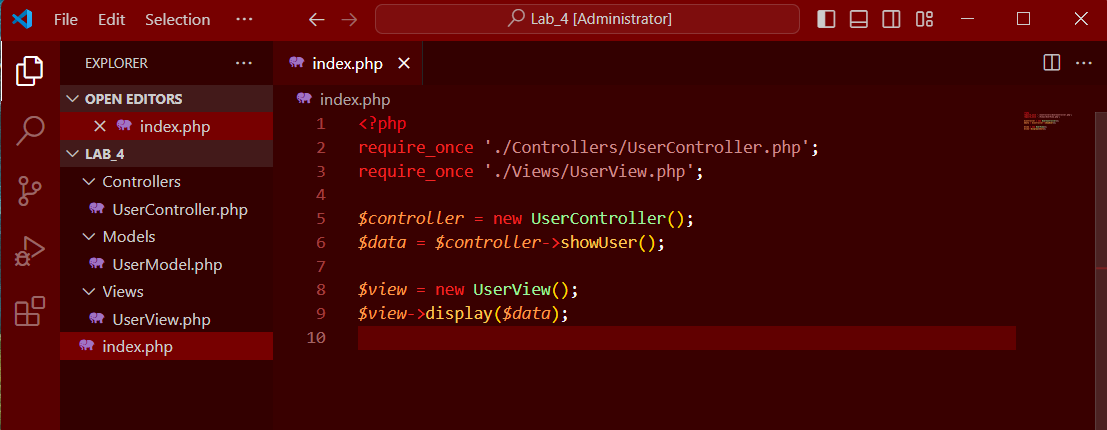
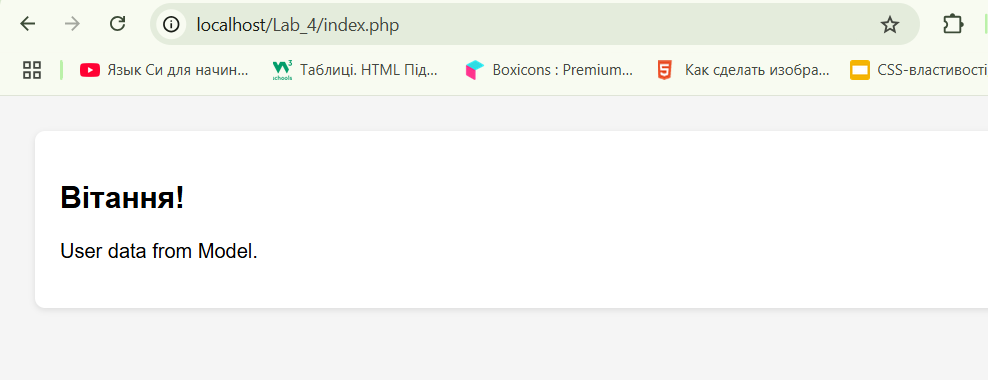
**Лабораторна робота №4**

**Тема: Об'єктно-орієнтоване програмування в РНР**

**Завдання 1. (**Організація класів по каталогах в проєкті)

* Створіть пустий проєкт PHP.
* Створіть каталоги: "Models", "Controllers", "Views".
* У кожному каталозі створіть по одному класу, наприклад, "UserModel", "UserController", "UserView".
* В кожному класі реалізуйте просту функціональність, наприклад, виведення повідомлення чи повернення значень.

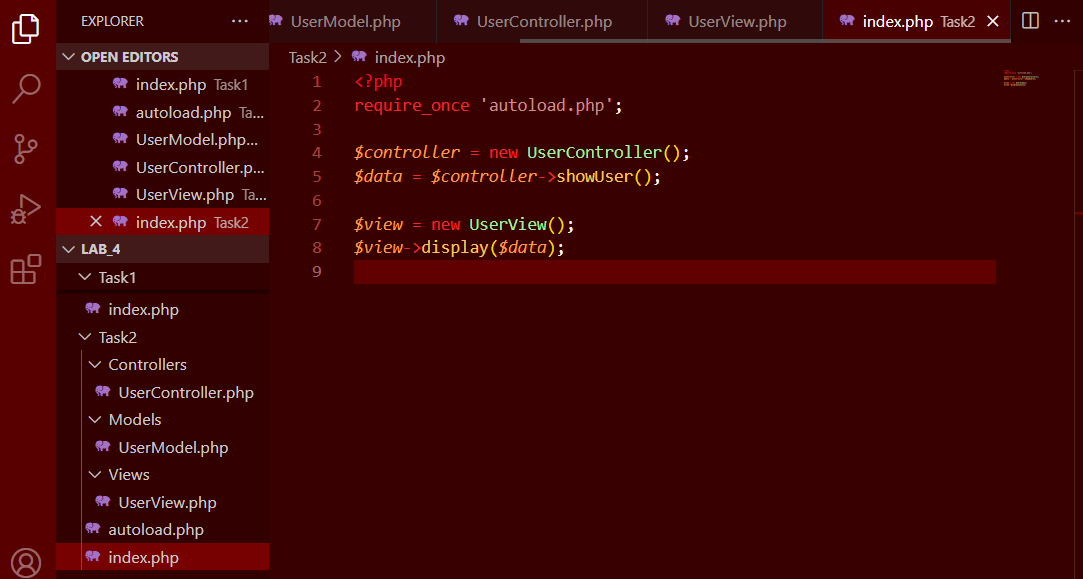


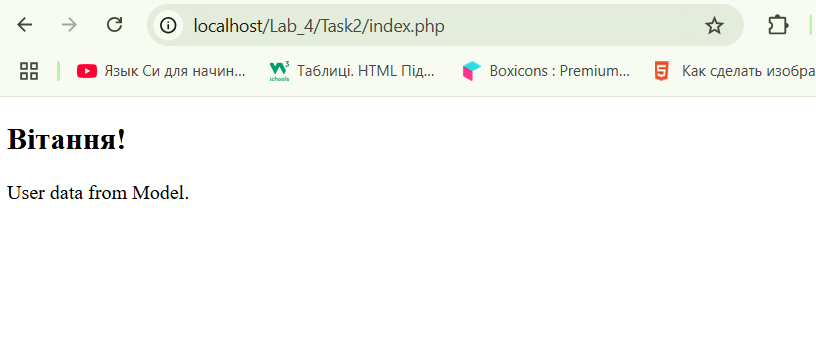


**Завдання 2. (**Автопідключення класів за допомогою spl\_autoload\_register. PHPDoc)

* Додайте PHPDoc коментарі до всіх класів, вказавши їх призначення та властивості.
* Створіть файл **autoload.php**, який буде містити функцію для автопідключення класів.

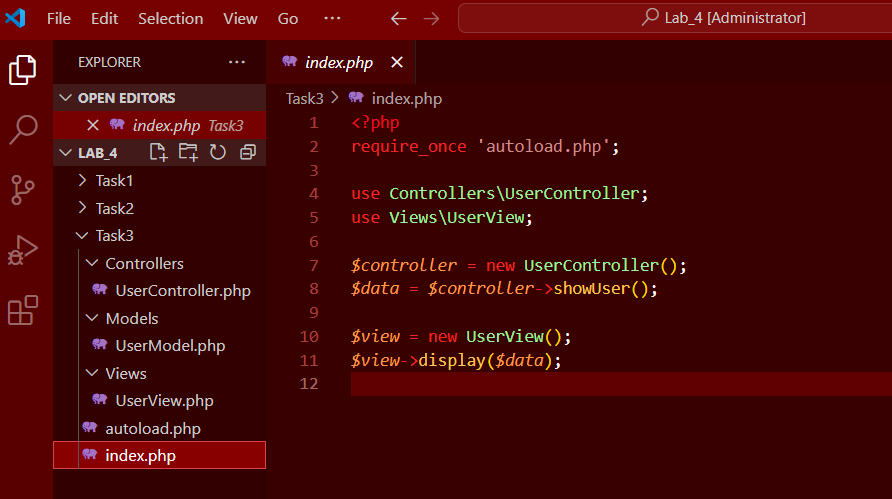
Використайте **spl\_autoload\_register** для автоматичного підключення класів на основі їхніх імен та розташування.

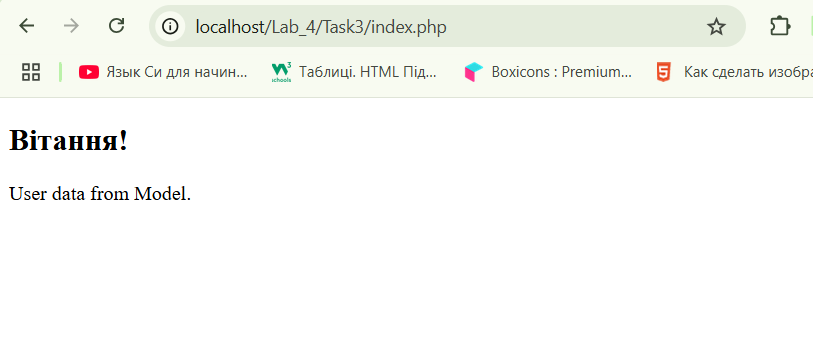




**Завдання 3.** (Неймспейси)

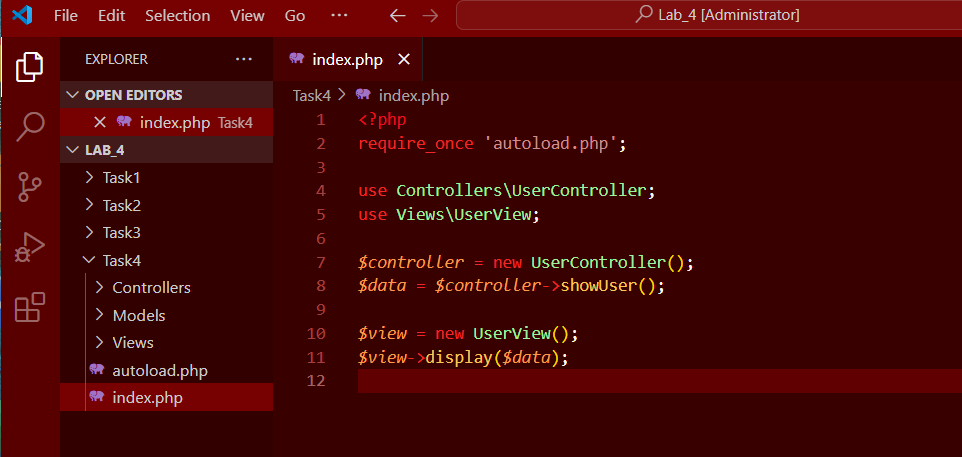
* Додайте неймспейси до класів у попередньому завданні. Наприклад, "namespace Models;" для "UserModel".
* Змініть файл **autoload.php** так, щоб він також враховував неймспейси при підключенні класів

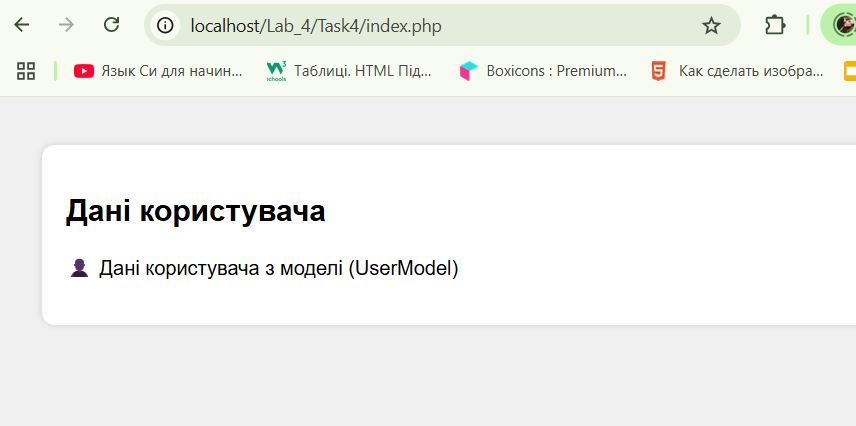




**Завдання 4. (**Автопідключення класів з неймспейсами)

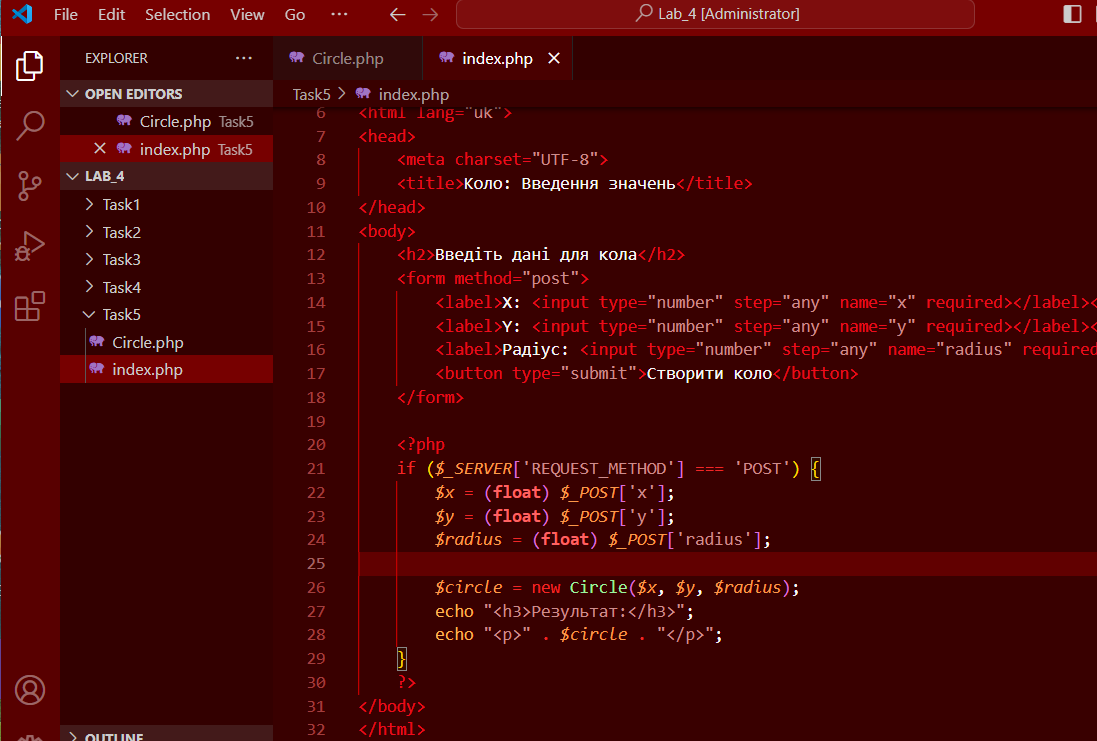
* Використовуйте аналогічний підхід до підключення класів, але тепер з урахуванням неймспейсів.
* Переконайтеся, що класи виводять повідомлення чи результати виклику.





**Завдання 5 (**Створення класу. Методи GET i SET)

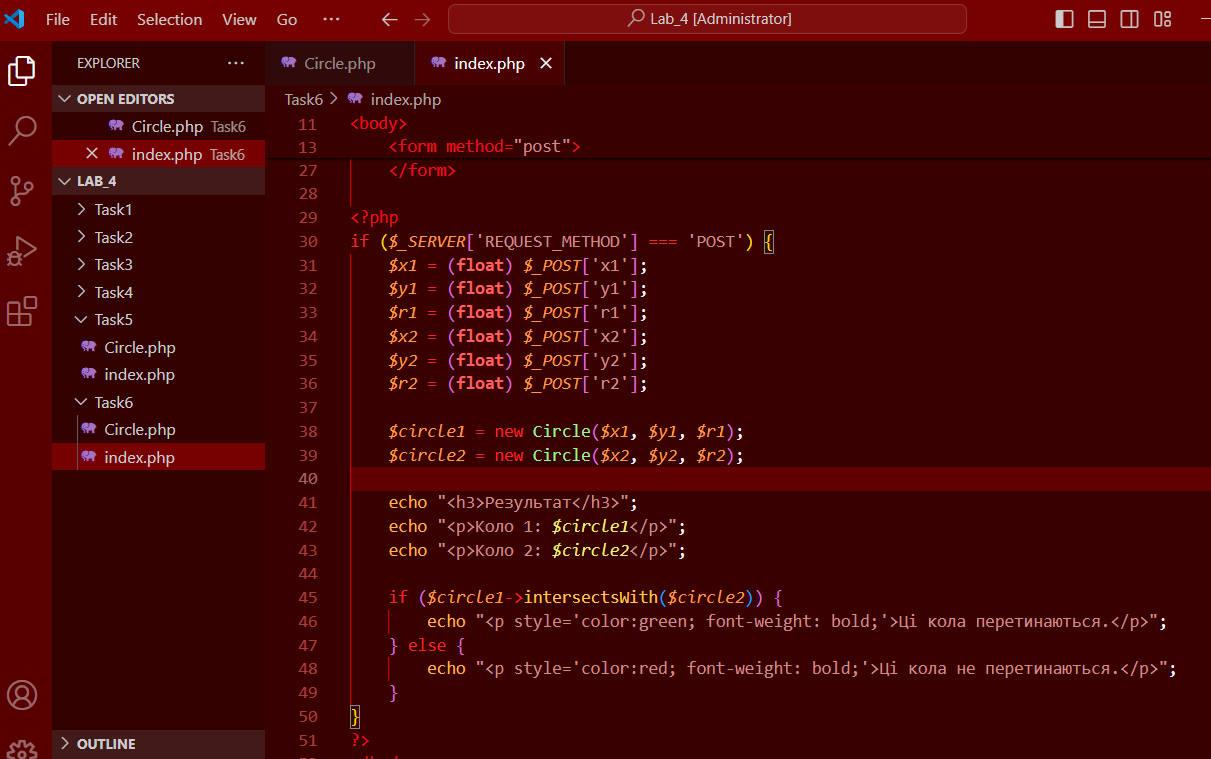
1. Створіть клас **Circle** з полями: координати центру і радіус кола
2. Створіть конструктор, що приймає значення для 3-х полів
3. Створіть метод **\_\_toString()**, що повертає рядок в форматі: «Коло з центром в (х, у) і радіусом radius»
4. Створіть методи **GET** і **SET** для всіх 3-х полів
5. Створіть об’єкт та перевірте всі його методи

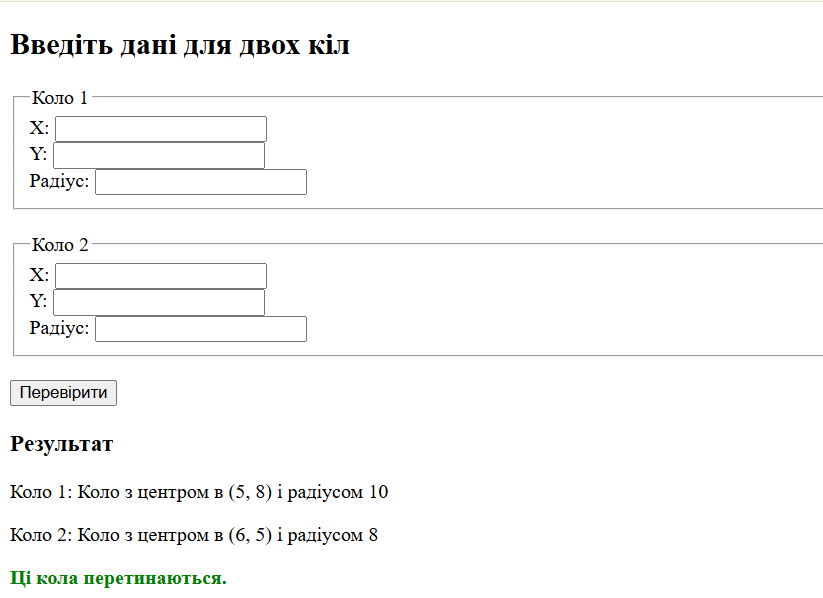


\

**Завдання 6** (Модифікатори доступу)

1. В класі з попереднього завдання зробіть всі поля private.
2. Створіть метод, що приймає об’єкт коло, і повертає **true**, якщо дані кола перетинаються, і **false**, якщо вони не перетинаються.





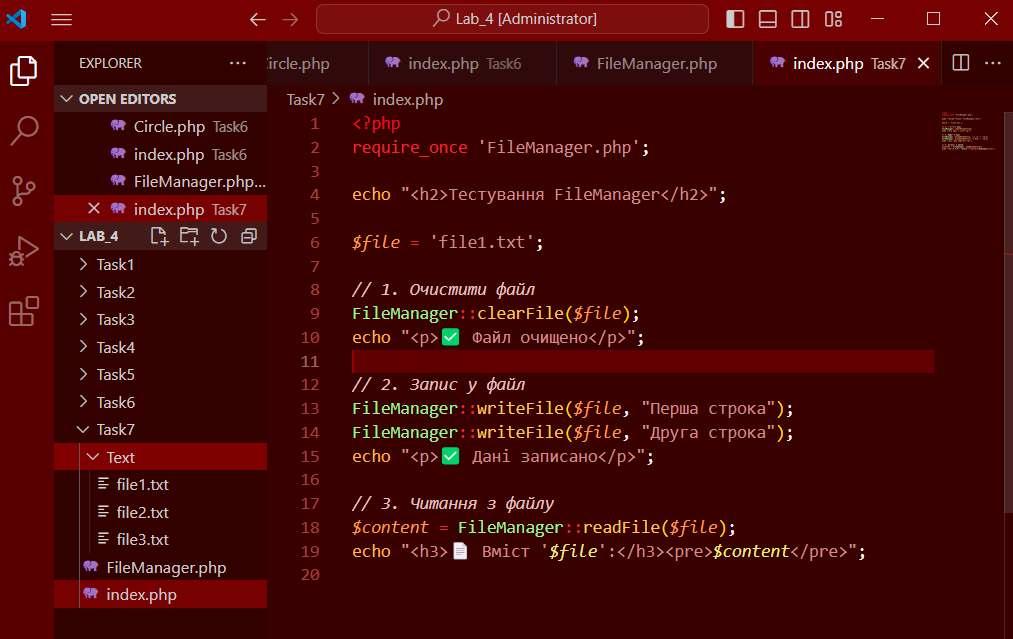
**Завдання 7 (**Статичні властивості і методи)

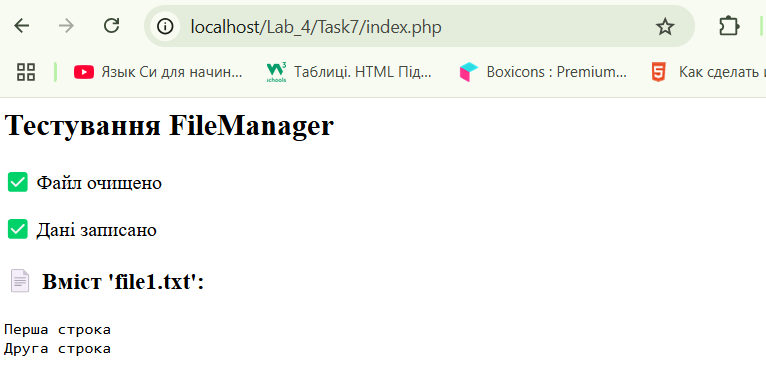
1. Створіть директорію **text**, а в ній 3 текстових файла
2. Створіть клас зі статичним полем **dir=”text”**
3. Створіть 2 статичних методи в класі: на читання та запис в файл:

* Ім’я файлу передається як параметр метода.
* В метод «**на запис в файл**» передається ще й рядок, який потрібно дописати в файл.
* Директорія береться зі статичного поля

     4) Створіть метод, що дозволяє стерти вміст файлу

Перевірте роботу всіх методів

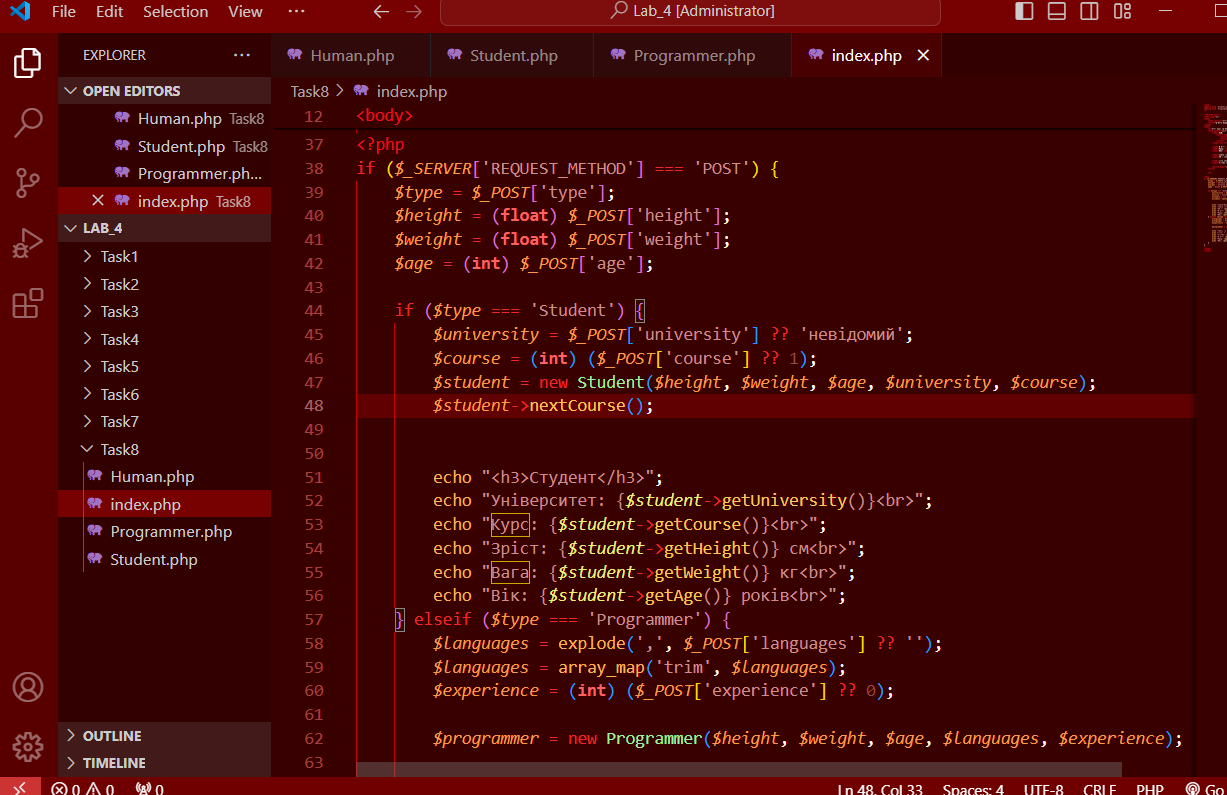


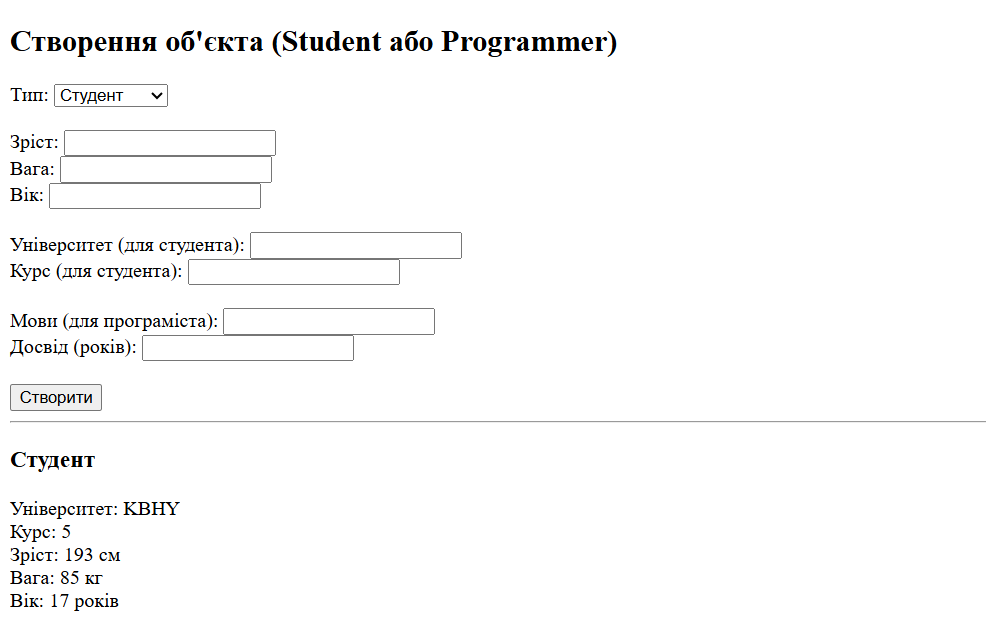


**Завдання 8** (Наслідування)

1. Створіть клас **Human** з властивостями, що характеризують людину (зріст, маса, вік…). Створіть методи **GET** i **SET** для кожної властивості
2. Створіть клас **Student**, який успадковуватиметься від класу **Human**:
3. Додайте властивості, специфічні тільки для студента (назва ВНЗ, курс…)
4. Додайте в клас методи **GET** i **SET** для всіх нових властивостей.
5. Реалізуйте метод, який буде переводити студента на новий курс (тобто просто збільшувати значення поля «курс» на 1)
6. Створіть клас **Programmer**, який успадковуватиметься від класу **Human**:

* Додайте властивості, специфічні тільки для програміста (масив з мовами програмування, які він знає, досвід роботи…).
* Додайте в клас методи **GET** i **SET** для всіх нових властивостей.

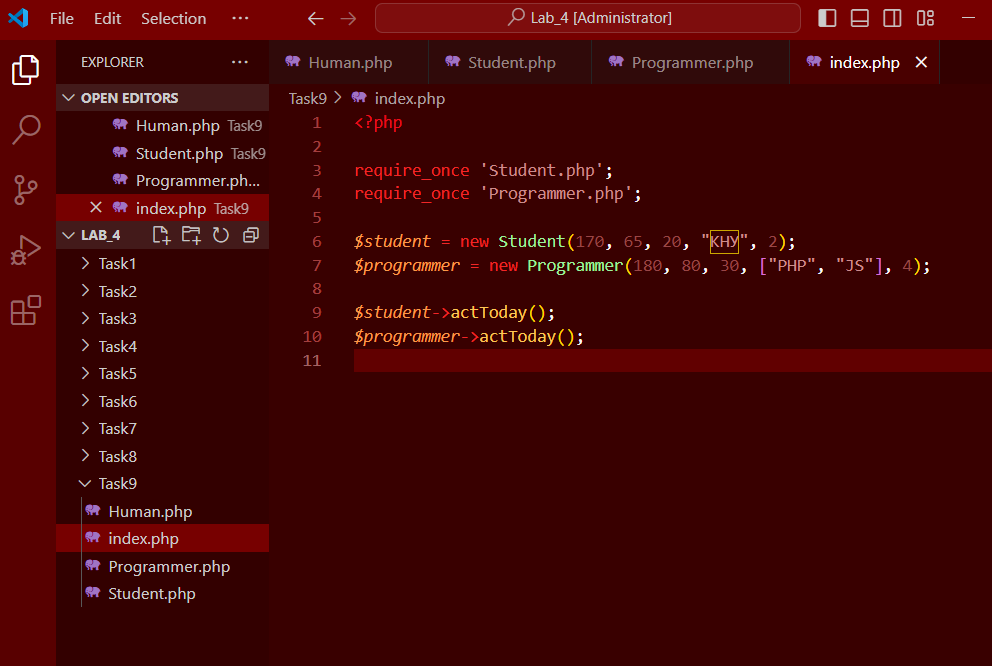
\

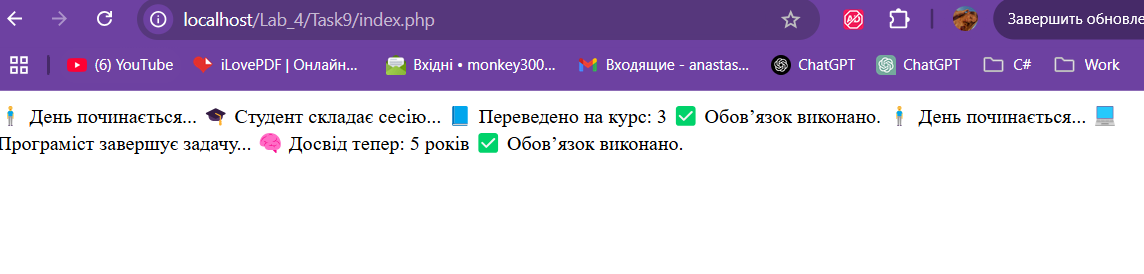


**Завдання 9 (**Абстрактні класи)

1. Зробіть клас **Human** абстрактним.
2. Напишіть метод «**Народження дитини**» в класі **Human**, що викликає метод «**Повідомлення при народженні дитини**» (не забудьте поставити модифікатор **protected**), який буде абстрактним
3. Реалізуйте «**Повідомлення при народженні дитини**» у класів **Student** та **Programmer**

Перевірте роботу методів «народження»





**Завдання 10 (**Інтерфейси)

1. Створіть інтерфейс «**Прибирання будинку**», в якому опишіть 2 методи: «**Прибирання кімнати**» і «**Прибирання кухні**»
2. Додайте створений інтерфейс в клас **Human**
3. Реалізуйте у кожному класі-спадкоємці (**Student** та **Programmer**) обидва методи
4. Реалізація повинна бути у вигляді одного з рядків: «**Студент прибирає кімнату**», «**Студент прибирає кухню**», «**Програміст прибирає кімнату**», «**Програміст прибирає кухню**»,
5. Перевірте роботу методів прибирання в обох класах

