

Курс: «Створення web-додатків за допомогою PHP, MySQL та Ajax»

Модуль 4. ООП, регулярні вирази

Тема: ООП. Часть 4

1. Розробити клас **Control** який описує базові атрибути стилю любого html-елементу та буде базою для кожного наступного класу, в якому створити **private** поля **background**, **width**, **height**, створити методи **getBackground()**, **setBackground()**, **getWidth()**, **setWidth()**, **getHeight()**, **setHeight()**. Методи, які починаються з **get** — мають повертати відповідне їм поле та не приймають параметрів, методи **set** — приймають параметр та записують його в відповідне поле.
2. Зробити клас **Input** який описує базові атрибути тегу **<input>**, унаслідувати клас **Control**. Додати до класу **Input** **private** поля **name**, **value** та методи **getName()**, **setName()**, **getValue()**, **setValue()**. Методи, які починаються з **get** — мають повертати відповідне їм поле та не приймають параметрів, методи **set** — приймають параметр та записують його в відповідне поле.
3. Зробити клас **Button**, який буде будувати на сторінці **button**, унаслідувати клас **Input**. Додати до класу **Button** **private** поле **isSubmit**, метод **getSubmitState()**, який не приймає параметрами нічого та повертає значення поля **isSubmit**, **setSubmitState()**, який записує в поле **isSubmit** значення **'true'**, та конструктор, який приймає параметрами **_background**, **_width**, **_height**, **_name**, **_value**, **_isSubmit** та викликає відповідні методи для заповнення параметрів у поля.

4. Зробити клас `Text`, який буде будувати на сторінці `textbox`, унаслідувати клас `Input`. Додати до класу `Text` `private` поле `placeholder`, метод `getPlaceholder()`, який не приймає параметри нічого та повертає значення поля `placeholder`, `setPlaceholder()`, який приймає параметром стрічку та записує її в поле `placeholder`, та конструктор, який приймає параметрами `_background`, `_width`, `_height`, `_name`, `_value`, `_placeholder` та викликає відповідні методи для заповнення параметрів у поля.
5. Зробити клас `Label`, який буде додавати `label` до усіх елементів керування, унаслідувати клас `Control`. Додати до класу `Label` `private` поле `for`, метод `getParentName()`, який не приймає параметри нічого та повертає значення поля `for`, `setParentName()`, який приймає параметром об'єкт класу `Button` або `Text`, викликає з під цього об'єкта метод `getName()` та записує в поле `for` результат його виконання, та конструктор, який приймає параметрами `_background`, `_width`, `_height`, `_name`, `_value`, `_forObject` та викликає відповідні методи для заповнення параметрів у поля.
6. В файлі `index.php` створити функцію `convertToHTML()`, яка на вхід приймає об'єкт класу `Control` та перетворює поля на HTML код та повертає його.
7. Створити об'єкт класу `Button`, та `Text`. Додати до об'єкта класу `Text` об'єкт класу `Label` та викликати для всіх них функцію `convertToHTML()`.

Приклад виводу:

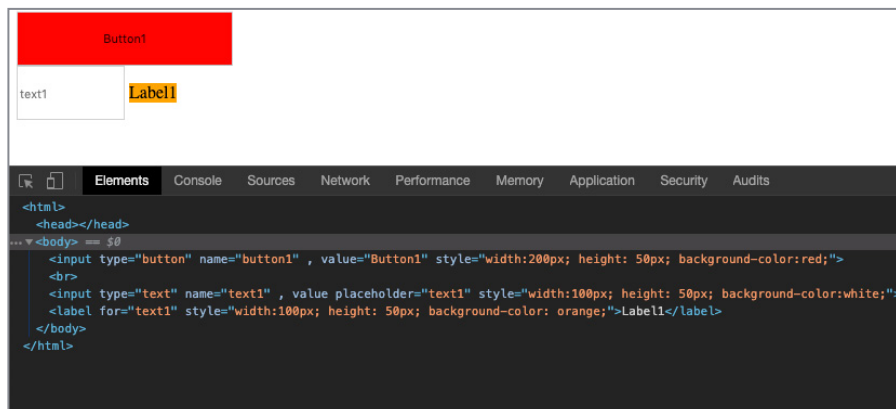


Рисунок 1