Класс FormHelper в ООП на PHP

Давайте теперь реализуем класс FormHelper, с помощью которого можно будет создавать формы. При этом унаследуем этот класс от класса TagHelper, созданного нами в предыдущем уроке.

Вот моя реализация класса FormHelper:

```php
<?php
  class FormHelper extends TagHelper
  {
    public function openForm($attrs = [])
    {
      return $this->open('form', $attrs);
    }

    public function closeForm()
    {
      return $this->close('form');
    }

    public function input($attrs = [])
    {
      if (isset($attrs['name'])) {
        $name = $attrs['name'];

        if (isset($_REQUEST[$name])) {
          $attrs['value'] = $_REQUEST[$name];
        }
      }

      return $this->open('input', $attrs);
    }

    public function password($attrs = [])
    {
      $attrs['type'] = 'password';
      return $this->input($attrs);
    }

    public function hidden($attrs = [])
    {
      $attrs['type'] = 'hidden';
      return $this->open('input', $attrs);
    }

    public function submit($attrs = [])
    {
      $attrs['type'] = 'submit';
      return $this->open('input', $attrs);
    }

    public function checkbox($attrs = [])
    {
      $attrs['type'] = 'checkbox';
      $attrs['value'] = 1;

      if (isset($attrs['name'])) {
        $name = $attrs['name'];

        if (isset($_REQUEST[$name]) and $_REQUEST[$na
```

```
            me] == 1) {
54            $attrs['checked'] = true;
55          }
56
57          $hidden = $this->hidden(['name' => $name, 'va
              lue' => '0']);
58        } else {
59          $hidden = '';
60        }
61
62        return $hidden . $this->open('input', $at
            trs);
63      }
64    }
65  ?>
```

Давайте применим наш класс для создания формы:

```
1   <?php
2     $fh = new FormHelper();
3
4     echo $fh->openForm(['action' => '', 'me
        thod' => 'GET']);
5       echo $fh->input(['name' => 'year']);
6       echo $fh->checkbox(['name' => 'check']);
7       echo $fh->submit();
8     echo $fh->closeForm();
9   ?>
```

В результате получится следующий HTML код:

```
1   <form action="" method="GET">
2     <input name="year">
3     <input type="hidden" name="check" value="0">
4     <input type="checkbox" name="check"
        value="1">
5     <input type="submit">
6   </form>
```

Задача 80.1

Изучите и разберите мой код описанного класса. Создайте с его помощью какую-нибудь HTML форму, применив как можно больше методов этого класса.

Задача 80.2

Самостоятельно, не подсматривая в мой код, реализуйте описанный класс.

Задача 80.3

Добавьте в вашу реализацию метод для создания тега textarea. Пусть этот тег сохраняет свое значение после отправки формы.

Задача 80.4

Добавьте в вашу реализацию метод для создания выпадающего списка. Пусть метод первым параметром принимает массив атрибутов тега select, а вторым - массив для создания тегов option. Пусть этот массив содержит ключ 'text' для текста пункта списка и ключ 'attrs' для массива атрибутов пункта списка.

Вот пример использования описанного метода:

```php
<?php
  $fh = new FormHelper();

  echo $fh->select(
    ['name' => 'list', 'class' => 'eee'],
    [
      ['text' => 'item1', 'attrs' => ['val
        ue' => '1']],
      ['text' => 'item2', 'attrs' => ['val
        ue' => '1', 'selected' => true]],
      ['text' => 'item1', 'attrs' => ['val
        ue' => '1', 'class' => 'last']],
     ],

  );
?>
```

В результате должен получится следующий HTML код:

```html
<select name="list" class="eee">
  <option value="1">item1</option>
  <option value="2" selected>item2</option>
  <option value="3" class="last">item3</option>
</form>
```

Сделайте так, чтобы выпадающий список сохранял свое значение после отправки формы.