**Практическая работа 5.** **Условные директивы и циклы в шаблонизаторе Blade**

**Инструкция:**

1. Продолжить выполнение в проекте \_фамилия-laravel
2. Выполнить приведенные ниже задания. Результаты выполнения поместить в отчет по практической работе в следующем виде:

* текст задания
* скрин вашего кода
* скрин результата в браузере (на скрине должна быть видна адресная строка)

**Задания для выполнения:**

Условия:

1. Пусть из действия в представление передается номер дня недели. Сделайте так, чтобы, если передан номер воскресенья, то на экран вывелся текст 'Воскресенье'. Пусть текст выводится в абзаце.
2. Модифицируйте предыдущую задачу так, чтобы на экран выводилось названия дня недели в том случае, если сегодня выходной день (то есть или суббота или воскресенье). Для решения задачи напишите две директивы **@if**.
3. Пусть из действия в представление передается номер дня недели. Сделайте так, чтобы, если номер дня недели соответствует выходному дню, то вывелось сообщение об этом, в противном случае выведите сообщение о том, что передан номер рабочего дня недели.
4. Пусть из действия в представление передается номер месяца. С помощью директив **@if** и **@elseif** определите пору года, в которую попадает переданный месяц. Если переданный номер не соответствует ни одному месяцу - выведите сообщение об этом.
5. Пусть из действия в представление передается возраст пользователя. Если возраст пользователь несовершеннолетний, то выведите сообщение об этом. Для решения задачи воспользуйтесь директивой **@unless**.
6. Пусть из действия в представление передается массив с числами. Если количество элементов в этом массиве больше нуля, то выведите на экран сумму этих элементов, а если количество элементов равно нулю, то выведите сообщение об этом.

Циклы:

1. Передайте из действия в представление массив с числами. Выведите этот массив в виде списка ul.
2. Модифицируйте предыдущую задачу так, чтобы в пункты списка выводились квадраты элементов массива.
3. Модифицируйте предыдущую задачу так, чтобы в пункты списка выводились квадратные корни элементов массива.
4. Передайте из действия в представление массив со строками. Выведите этот массив в виде списка **ul** так, чтобы выводились и ключи, и значения элементов массива.
5. Модифицируйте предыдущую задачу так, чтобы на выводе к ключу прибавлялась единица.
6. Передайте из действия в представление массив с числами. Выведите этот массив в виде списка **ul**. Сделайте так, чтобы в список попадали только элементы, значениями которых служат четные числа.
7. Из действия в представление передается переменная **$data**. Эта переменная может содержать массив с числами либо быть обычным числом. Если переменная **$data** - массив, то выведите элементы этого массива в виде списка **ul**, а если переменная **$data** - число, то выведите это число в абзаце.
8. Передайте из действия в представление массив с числами от 1 до 25. Выведите его в виде таблицы 5 на 5.
9. Передайте из действия в представление массив с работниками:

$employees = [

   [

**'name'** => **'user1'**,

**'surname'** => **'surname1'**,

**'salary'** => 1000,

   ],

   [

**'name'** => **'user2'**,

**'surname'** => **'surname2'**,

**'salary'** => 2000,

   ],

   [

**'name'** => **'user3'**,

**'surname'** => **'surname3'**,

**'salary'** => 3000,

   ],

];

Выведите этот массив в виде списка **ul**.

10. Выведите массив из предыдущей задачи в виде таблицы **table**.

11.Из действия в представление передается массив со строками. Выведите элементы этого массива в виде списка **ul**. Сделайте так, чтобы перед значением элемента, выводился еще и порядковый номер итерации, начиная с 0.

12. Модифицируйте предыдущую задачу так, чтобы порядковый номер итерации начинался с 1.

13. Модифицируйте предыдущую задачу так, чтобы для первой **li** добавлялся CSS класс **first**, а для последней - класс **last**.

14. Из действия в представление передается массив с числами. Выведите каждый элемент этого массива в своем теге **b**. С помощью **$loop->remaining** три последних элемента массива выведите не в теге **b**, а в теге **i**.

15. Воспользуйтесь примером работы с циклами выше и проверьте работу директив @break, @continue, а также директивы **@forelse**

16. С помощью цикла **$@for** выведите 10 абзацев, заполненных числами от 1 до 10.