**HTML Elements Page**

«Типичные операции в PHP с использованием ООП»

Особое внимание уделяйте устойчивости работы вашего скрипта (т.е. предусматривайте возможные ошибочные ситуации и добавляйте реакцию на них).

**Задание**

Напишите PHP-скрипты (по одному файлу на каждый из приведённых ниже пунктов):

1. Создайте класс FormBuilder, который будет позволять формировать HTML-форму. Например, такое использование вашего класса

$formBuilder = new FormBuilder(FormBuilder::*METHOD\_POST*, '/destination.php', 'Send!');  
$formBuilder->addTextField('someName', 'Default value');  
$formBuilder->addRadioGroup('someRadioName', ['A', 'B']);  
$formBuilder->getForm();

должно приводить к генерации такого HTML-кода:

<*form* method="post" target="/destination.php">  
 <*input* type="text" name="someName" value="Default value" />  
 <*input* type="radio" name="someRadioName" value="A" />  
 <*input* type="radio" name="someRadioName" value="B" />  
 <*input* type="submit" value="Send!" />  
</*form*>

1. Создайте класс SafeFormBuilder, унаследованный от класса FormBuilder из первого задания, который будет анализировать суперглобальные массивы $\_POST и $\_GET и (если там есть данные, полученные при предыдущей отправки формы) формировать значения по умолчанию для соответствующих полей. Например, если в вашей форме есть поле <*input* type="text" name="someName" />, и в суперглобальном массиве $\_POST есть элемент someName со значением «ABC», то ваш класс должен генерировать такой код: <*input* type="text" name="someName" value="ABC" />.
2. Создайте класс Logger, который будет при создании объекта позволять указывать, выводить ли сообщение на экран или в файл, а также будет добавлять к каждому сообщению (в начало) дату и время.
3. Создайте класс SmartDate, который бы при создании объекта получал значение даты и для этой даты мог определять: является ли этот день выходным (СБ, ВС); расстояние в указанных единицах времени между этим днём и сегодняшним днём; является ли год, к которому относится дата, високосным.
4. Создайте класс CryptoManager, который позволял бы шифровать и дешифровывать переданный текст; алгоритм шифрования и пароль должны передаваться в конструктор.

**Дополнительные задания (для желающих потренироваться)**

Напишите PHP-скрипты (по одному файлу на каждый из приведённых ниже пунктов):

1. По аналогии с классом FormBuilder создайте класс TableBuilder для автоматизации формирования HTML-таблиц.
2. Создайте класс FileSystemObject, который позволял бы хранить имя, размер, тип объекта файловой системы, а также возвращать размер в любых указанных единицах (B, KB, MG, GB и т.д.). С использованием полученного класса прочитайте содержимое текущего каталога и выведите все файлы с их размерами в MB.

Форма отчётности: набор файлов, названных в соответствии с номером задания, например «task\_01.php», «task\_02.php», «task\_additional\_01.php», «task\_additional\_02.php» и т.д.