# Лабораторная работа 4

1. Создайте в проекте директорию include. В этой директории создайте

**#5** файлы common.inc.php, string.inc.php, request.inc.php

Подключите файлы string.inc.php и request.inc.php в файле common.inc.php

2. Разработайте функцию last($str).

**#10** Функция last принимает строку символов и возвращает лишь последний символ

из этой строки.

Функция должна размещаться в файле string.inc.php

Для проверки функции разработайте программу /last.php

В GET-параметре str передается значение исходной строки.

Необходимо учесть, что 1) параметр может быть не передан,

2) строка может быть пустой.

3. Разработайте функцию withoutLast($str).

**#10** Функция withoutLast принимает строку и возвращает ту же самую строку

без последнего символа.

Функция должна размещаться в файле string.inc.php

Для проверки функции разработайте программу without\_last.php

4. Разработайте функцию revers($str).

**#10** Функция принимает строку в качестве параметра.

Возвращает строку, состоящую из тех же символов, что и исходная строка, но

символы идут в обратном порядке.

Для решения этой задачи не используйте встроенные в язык PHP функции работы со строками (такие как reverse например)

Для проверки функции разработайте программу reverse.php

5. Разработайте PHP приложение Check Identifier, которое проверят является ли

**#10** переданный в GET параметр identifier индетификатором по правилу SR3.

Программа должна выводить yes или no, а также поясняющую информацию в том случае, если

передананя строка не является идентификатором.

**Задание должно быть выполнено без использования регулярных выражений.**

## Дополнительное задание

6. Разработайте приложение Remove Extra Blanks на языке PHP.

**#20** В запросе GET передается параметр text.

Скрипт выводит в стандартный поток вывода этот же текст без лишних пробелов.

Результирующая строка не должна содержать пробелов в начале и в конце строки.

Внутри строки не может быть более одного пробела подряд.

7. Разработайте программу для проверки надежности пароля Password Strength.

**#30** В GET параметре password передается пароль для анализа. Пароль может состоять только из английских

символов в верхнем и нижнем регистрах, а также из цифр.

Надежность пароля вычисляется по следующему принципу.

1) Изначально считаем надежность равной 0.

2) К надежности прибавляется (4\*n), где n - количество всех симоволов пароля

3) К надежности прибавляется +(n\*4), где n - количество цифр в пароле

4) К надежности прибавляется +((len-n)\*2) в случае, если пароль содержит n символов в верхнем регистре

5) К надежности прибавляется +((len-n)\*2) в случае, если пароль содержит n символов в нижнем регистре

6) Если пароль состоит только из букв вычитаем число равное количеству символов.

7) Если пароль состоит только из цифр вычитаем число равное количеству символов.

8) За каждый повторяющийся символ в пароле вычитается количество повторяющихся символов

Например: abcd1a

вычитаем -2 поскольку символ a встречается дважды.

Программа должна выводить на экран надежность пароля в виде числа.

8. Необходимо разработать небольшой проект, состоящий из главного PHP файла,

**#30** являющегося точкой входа в приложение, директории inc, где лежит произвольный набор подключаемых файлов и скрипта-автозагрузчика. Каждый из подключаемых файлов директории inc содержит обычный вывод с помощью функции echo.

Пример структуры проекта:

inc

| |\_ autoload.inc.php

| |\_ hello.inc.php - echo “Hello”;

| |\_ world.inc.php - echo “World”;

|

|\_ main.php

При добавлении новых PHP-файлов в директорию .inc происходит их автоматическое подключение, что можно проверить, запустив скрипт main.php. В приведенном выше примере будет выведена строка “HelloWorld”.