

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления» Кафедра «Системы обработки  
информации и управления»



**Студент Андреев К.А.**

**Отчет**

**по выполнению лабораторной работы По курсу**

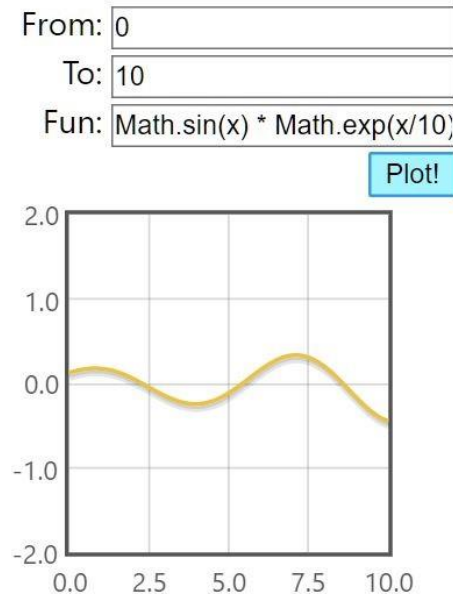
**“Разработка интернет-приложений”**

**Лабораторная работа № 7**

## Задание и порядок выполнения

Разработать приложение для построения графиков тригонометрических функций на языке Javascript с HTML интерфейсом.

Пример интерфейса:



Ход работы:

1. Ознакомиться с теоретической частью
2. Создайте новый проект PyCharm тип проекта: Pure Python  
(мы не будем использовать Python в этой работе, просто это позволяет создать абсолютно пустой проект без зависимостей)
3. Добавьте в проект 2 файла:
  - a. index.html
  - b. index.js
4. Сверстайте страницу со следующими элементами:
  - a. два поля ввода для области определения аргумента (`<input>`)
  - b. поле для ввода функции (`<input>`)
  - c. кнопка "Построить график" (`<button>`)
  - d. поле вывода графика (`<div>`)
5. При помощи css укажите размеры блока графика, отличные от нуля
6. Присвойте каждому полю уникальный class (например, from, to, fun, output и т.д.)
7. Убедитесь, что ваша страница отображается в браузере нормально
8. Подключите jQuery, flot и ваш скрипт в index.html, используя теги `<script>`  
`<script src="https://code.jquery.com/jquery-2.2.4.min.js">< /script>`  
`<script src="https://cdnjs.cloudflare.com/ajax/libs/flot/0.8.3/jquery.flot.js">< /script>`
9. Переходим к разработке скрипта
10. Дождитесь загрузки страницы

```
$(function() {  
    // ...  
})
```

11. Найдите все элементы управления на вашей странице `var $from = $('from');`
12. Подпишитесь на событие нажатия кнопки  
`$button.click(onClick);`
13. Отмените действие по-умолчанию (отправку формы)  
`e.preventDefault()`
14. Получите значения из полей ввода  
`$from.val()`
15. Не забудьте преобразовать числовые значения из строк в числа `parseFloat`, `parseInt`
16. Создайте массив пар значений  
`const points = [[x1, y1], ..., [xn, yn]];`
17. Для того, чтобы получить значение функции, заданной в виде строки, используйте функцию `eval()`  
`const x = 0.1; const fun =  
 'Math.sin(x)'; const  
  
 y = eval(fun);`
18. Постройте график по точкам  
`$.plot($output, [ points ], {});`
19. Проверьте правильность работы приложения, в случае проблем, воспользуйтесь отладчиком Chrome DevTools
20. Проверьте построение графиков функций:
  - a. `Math.sin(x)`
  - b. `Math.random()`
  - c. `Math.exp(x)`
21. Выведите название построенной функции в легенду:  
<http://www.flotcharts.org/plot/examples/basic-options/index.html>
22. Дополнительное задание: сделайте анимацию графика функции как на осциллографе для этого по таймеру `setInterval()` / `clearInterval()` перестраивайте график функции, прибавляя к **x** изменяющийся коэффициент **dx**

## Исходники:

### Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Построитель графиков</title>
  <script src="https://code.jquery.com/jquery-2.2.4.min.js"></script>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/flot/0.8.3/jquery.flot.js"></script>
  <script type="text/javascript" src="index.js"></script>
  <style>
    body{
      margin-left: 0px;
      margin-top: 0px;
      background-color: wheat;
    }
  </style>
</head>
<body>

<h1 style="margin-left: 0px; margin-top: 0px; background-color: coral; font-
family: Verdana; text-align: center;">Построитель графиков</h1>
<div style="margin-left: 10px">
  <div>Границы отображения</div>
  <input type="text" class="from" size="10" value="-10" style="text-align:
center"></input> | <input type="text" class="to" size="10" value="10"
style="text-align: center"></input><br>
  Введите функцию<br>
  <input type="text" class="fun" style="text-align: center"></input> <br>
  <label><input type="checkbox" id="iddd" />Осциллограф?</label><br>
  <button style="color: red; margin-top: 10px; height: 30px">Построить
график</button><br>
  <div class="plot1" style="width: 400px; height: 200px; margin-top:
5px"></div>

</div>

</body>
</html>
```

### Index.js

```
$(function() {
  var $from = $('.from');
  var $to = $('.to');
  var $fun = $('.fun');
  var $button = $('.button');
  var $plot1 = $('.plot1');
  var check;
  const $dx = 0.05
  check=document.getElementById('iddd');

  $button.click(function(e) {
    e.preventDefault();
    $funn = $fun.val()
    if ($funn!=0) {

      $fromm = parseFloat($from.val());
      $too = parseFloat($to.val());
```

```

var $values = []
for (var x = $fromm; x <= $too; x += $dx) {
    $values.push([x, eval($funn)])
}

$.plot($plot1, [{label: $funn, data: $values}], {})
if (check.checked){
    karl = setInterval(function(){
        $fromm += $dx;
        $too += $dx;
        $values = []

        for(var x = $fromm; x <= $too; x += $dx){
            $values.push([x, eval($funn)])
        }

        $.plot($plot1, [{label: $funn, data: $values}], {})
    }, 50);} else {clearInterval(karl);}

} else
{
    {alert("Вы не ввели функцию")}
}

});
})

```

## Результат:

