

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ВятГУ)

Факультет компьютерных и физико-математических наук
Кафедра фундаментальной математики

Выпускная квалификационная работа

по направлению подготовки

02.03.01 Математика и компьютерные науки

Профиль: «Математические основы компьютерных наук»

на тему «Разработка модуля интеграции онлайн-сервиса
построения графиков и геометрических чертежей в визуальный
онлайн HTML-редактор»

Выполнил: студент МКб-4301-51-00
Мельков Алексей Константинович

(подпись)

Научный руководитель:
канд. физ.-матем. наук, доцент
Марков Роман Владимирович

(подпись)

Допущено к защите в ГЭК: «___» _____ 20__ г.
Заведующий кафедрой фундаментальной математики

(подпись)

Е. М. Вечтомов

Декан факультета компьютерных и физико-математических наук

(подпись)

Н. А. Бушмелева

Киров
2022

Реферат

Мельков Алексей Константинович

Разработка модуля интеграции онлайн-сервиса построения графиков и геометрических чертежей в визуальный онлайн HTML-редактор

ПЗ : Выпускная квалификационная работа, каф. фундаментальной математики; рук. Марков Роман Владимирович. Киров, 2022.

ПЗ 16 с., 1 рис., 1 табл., 2 источников, 1 прил.

<КЛЮЧЕВЫЕ СЛОВА>

Объект разработки: модуль интеграции онлайн-сервиса построения графиков и геометрических чертежей в визуальный онлайн HTML-редактор

Цель: <Переносится цель ВКР дословно>

Методы проведения работы: <Описываются применяемые методы: анализ научной литературы, сравнительный анализ, математическое моделирование, методы математического анализа и т. д.>

Рассматриваются вопросы: связи онлайн-сервиса построения графиков и геометрических чертежей GeoGebra и визуальный онлайн HTML-редактор TinyMCE

Оглавление

Введение	3
ГЛАВА 1. Теоретическая часть	5
1.1. Анализ готовых решений	5
1.2. Описание своего решения	5
1.3. Необходимые знания	5
1.3.1 основы JavaScript	5
1.3.2 API MOODLE для плагинов.	8
1.3.3 API TinyMce	9
1.3.4 API Geogebra	10
1.4. Создание плагина для TinyMCE на платформе Moodle	10
ГЛАВА 2. Однако, глава	13
2.1. и раз	13
2.1.1 и два	13
Заключение	14
Библиографический список	15
ПРИЛОЖЕНИЕ А. Листинг программы	16

Введение

Выпускная квалификационная работа посвящена разработке модуля интеграции онлайн-сервиса построения графиков и геометрических чертежей в визуальный онлайн HTML-редактор

Актуальность работы обусловлена оптимизацией времени и удобства написания статей, вопросов, заданий. Данный модуль позволит интегрировать распространенный онлайн-сервис построения графиков и геометрических чертежей GeoGebra в онлайн HTML-редактор TinyMCE, который также присутствует на платформе Moodle. Что добавит возможность составления новых тестов, лекций, практических заданий для обучения студентов.

Объектом исследования является изучение связи программ TinyMCE, GeoGebra, платформы Moodle. Для этого потребовалось рассмотреть их API и построение плагина

Предмет исследования —

Цель работы: написание плагина для платформ Moodle, который расширяет возможности TinyMCE добавлением новой кнопки, которая сможет открыть сервис построения графиков GeoGebra.

Для достижения поставленной цели сформулированы следующие **задачи:**

1. Изучение основ программирования на языке JavaScript
2. Изучение API GeoGebra, TinyMCE
3. Узнать как создается плагин для расширения TinyMCE на платформе Moodle
4. Создание плагина

Теоретическая значимость работы состоит в следующем...

Практическая значимость работы состоит в следующем...

В целом работа носит **теоретический (или прикладной, или практический)** характер.

Структура работы. Выпускная квалификационная работа, общим объемом 16 стр., состоит из введения, двух глав, заключения, библиографического списка.

Первая глава посвящена теоритической части

Вторая глава посвящена практической части

В заключении представлены основные результаты дипломной работы.

В библиографический список включено (кол-во) источников.

Результаты работы **апробированы** (в докладах..., статьях ..., внедрены...)

ГЛАВА 1

Теоретическая часть

1.1 Анализ готовых решений

Я нашел в интернете несколько способов вставки графиков в текстовый редактор в Moodle. Первый способ представлен самой GeoGebra. Они предлагают с своего сайта построить необходимые графики, затем нажать комбинацию клавиш, что позволит скопировать HTML код. После этого в редакторе TinyMCE есть кнопка редактирования HTML кода, при ее нажатии откроется окно, куда надо вставить скопированные данные.

1.2 Описание своего решения

Я предлагаю расширить возможности редактора TinyMCE с помощью разработки собственной кнопки, которая при нажатии открывает окно GeoGebra. В этом окне можно построить графики, после чего нажать на внутреннюю кнопку вставки. В результате, все данные перенесутся в текстовую область. Полученный график будет виден при просмотре страницы.

1.3 Необходимые знания

1.3.1 основы JavaScript

JavaScript - это язык программирования. Поддерживает объектно-ориентированный и стили.

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Переменные. Они бывают разных типов:

— Number: любое число, включая десятичные дроби

- String: любая группа символов на вашей клавиатуре, заключенная в одинарные кавычки: '...' или двойные кавычки "...".
- Boolean: этот тип данных имеет только два возможных значения — либо true, либо false. Полезно думать о логических значениях как о переключателях включения и выключения или как об ответах на вопрос “да” или “нет”
- Null: этот тип данных представляет намеренное отсутствие значения и представлен ключевым словом null
- Undefined: этот тип данных обозначается ключевым словом undefined. Он также представляет отсутствие значения, хотя его использование отличается от null. Отличие в том что Undefined означает, что данное значение не существует.
- Object: коллекция связанных данных

Свойства. Когда вы вводите новый фрагмент данных в программу JavaScript, браузер сохраняет его как экземпляр типа данных. Все типы данных имеют доступ к определенным свойствам, которые передаются каждому экземпляру. Например, каждый экземпляр string имеет свойство length, в котором хранится количество символов в этой строке. Вы можете получить информацию о свойстве, добавив строку с точкой и именем свойства, к примеру, `console.log("Привет".length)` выведет в консоль 6

Методы - это действия, которые мы можем выполнять. Типы данных имеют доступ к определенным методам, которые позволяют нам обрабатывать экземпляры этого типа данных. Чтобы вызвать метод, необходимо после переменной написать точку, его название, парные скобочки (). К примеру, `console.log('привет'.toUpperCase())` выведет в консоль ПРИВЕТ

Функции. В программировании часто используется код для многократного выполнения определенной задачи. Вместо того, чтобы переписывать один и тот же код, можно сгруппировать блок кода вместе и связать его с одной задачей, затем можно повторно использовать этот блок кода всякий раз, когда нужно выполнить задачу снова. Мы достигаем этого, создавая функцию. Функция - это повторно используемый блок кода, который группирует последовательность инструкций для выполнения определенной задачи.

Как объявляется функция:

```
function Идентификатор(){  
    тело функции;  
}
```

Альтернативным способом можно внести функцию в переменную. Тогда синтаксис объявления поменяется следующим образом:

```
let Переменная = (параметры) => {  
    Тело_функции  
}
```

Объекты. Объекты JavaScript нужны для создания более сложных структур данных. По своей сути объекты JavaScript представляют собой контейнеры, хранящие связанные данные и функциональные возможности.

Создание объектов, могут быть назначены переменным точно так же, как и любому типу JavaScript. Для обозначения объекта используются фигурные скобки `{}`. Объект можно заполнять неупорядоченными данными. Эти данные организованы в пары ключ-значение. Ключ подобен имени переменной, которое указывает на место в памяти, содержащее значение. Значение ключа может иметь любой тип данных в языке, включая функции или другие объекты.

Мы создаем пару ключ-значение, записывая имя ключа или идентификатор, за которым следует двоеточие, а затем значение. Мы разделяем каждую пару ключ-значение в объектном литерале запятой `,:`

```
let Переменная_объекта = {  
    ключ: 'что-то',  
    'ключ': 123  
};
```

Есть два способа получить доступ к свойству объекта. Первый, как и лю-

бое свойство - через точку . и название самого свойства. Второй способ получить доступ к значению ключа - это использовать обозначение в квадратных скобках, [].

Для того, чтобы интегрировать JavaScript код в HTML страницу надо вставить парный тег <script>. Но также можно подключить JS файл с помощью ссылки на него тем же тегом, использованием атрибута src, то есть надо указать где лежит этот файл.

1.3.2 API MOODLE для плагинов.

API (Application programming interface) - это инструмент для взаимодействия нескольких программ. API содержит в себе некие «мостики», позволяющие программе А получить доступ к данным из программы Б или к некоторым ее возможностям. Таким образом, программисты могут расширять функциональность своего продукта и связывать его с чужими разработками. Для создания своего плагина TinyMCE необходимо добавить три php файла, которые обращаются к Moodle.

version.php содержит информацию о версии плагина Moodle

```
<?php
defined('MOODLE_INTERNAL') || (die);
#текущая версия плагина (Дата: ГГГГММДДЧЧ)
$plugin -> version = 2012112900;
#требующиеся версия Moodle (Дата: ГГГГММДДЧЧ)
$plugin -> requires = 2012112900;
#полное имя плагина
$plugin -> component = 'tinymce_ИмяПлагина';
```

lib.php содержит код плагина Moodle, здесь должен находиться по крайней мере класс tinymce_ИмяПлагина с методом update_init_params()

```
<?php
defined('MOODLE_INTERNAL') || (die);

class tinymce_ИмяПлагина extends editor_tinymce_plugin {
```

```
protected $buttons = array ('ИмяПлагина');

protected function update_init_params(array & $params, context $context,
array $options = null) {

    $this->add_button_after($params,3,'ИмяПлагина','spellchecker');

    $this->add_js_plugin($params);

}
}
```

tinymce_ИмяПлагина.php. Файл обязан содержать запись

```
<?php
$string['pluginname'] = 'ИмяПлагина';
```

где 'pluginname' нельзя изменять.

1.3.3 API TinyMce

Чтобы вставить TinyMCE редактор в HTML страницу надо добавить следующие скрипты

```
<script src="https://cdn.tiny.cloud/1/no-api-key/tinymce/5/tinymce.min.js"referrerp
```

Эта строка ссылает нашу страницу на сайт редактора

```
<script>
tinymce.init({
});
</script>
```

Этот скрипт запускает TinyMCE. Так как в Moodle уже загружен редактор и я расширяю его возможности, то эти скрипты мне не нужны. В итоге я изменяю параметры init.

1.3.4 API Geogebra

Чтобы на своей странице загрузить GeoGebra нужна добавить несколько скриптов и место куда будет добавляться приложение. Сначала нам нужно, сделать ссылку на источник GeoGebra: `<script src="https://www.geogebra.org/apps/deployggb.js"></script>`.

Затем создадим контейнер, куда вставим наше приложение: `<div id="ggb-element"></div>`

Теперь нам нужно задать параметры и открыть окно GeoGebra:

```
<script>
```

```
var params = {"appName": "graphing" "width": 800, "height": 600,
"showToolBar": true, "showAlgebraInput": true, "showMenuBar": true };
var applet = new GGBApplet(params, true);
window.addEventListener("load function() {
applet.inject('ggb-element');
});
```

```
</script>
```

Разберем параметры:

- `appName` - тип вычислений. Можно выбрать `graphing` (построение графиков), `geometry` (геометрия), `3d`.
- `width`, `height` - высота и ширина окна в px.
- `showToolBar`, `showAlgebraInput`, `showMenuBar` - отображение инструментов и панелей. Если `true` - отображаются, если `false` - скрыты

После мы создаем `GGBApplet` с указанными параметрами, при загрузке страницы, срабатывает событие добавления `GGBApplet` в `div` контейнер с `id 'ggb-element'`

1.4 Создание плагина для TinyMCE на платформе Moodle

Для создания плагина TinyMCE нужно сгенерировать каталог.

В нем минимум должны находиться:

- Подкаталог `/lang`
- Подкаталог `/pix`

- Подкаталог /tinymce
- Файл lib.php
- Файл version.php

/lang предназначен для хранения языковых папок. В моем примере в ней существует подкаталог /en в котором находится файл tinymce_ИмяПлагина.php

Файл обязан содержать запись

```
<?php
$string['pluginname'] = 'ИмяПлагина';
```

где 'pluginname' нельзя изменять.

/pix содержит иконку вашего плагина в расширении .png, обычно это 20 × 20 пикселей.

/tinymce основной каталог, содержащий js файл. В нем находится скрипт, что будет делать наш плагин.

lib.php содержит код плагина Moodle, здесь должен находиться по крайней мере класс tinymce_ИмяПлагина с методом update_init_params()

```
<?php
defined('MOODLE_INTERNAL') || (die);
```

```
class tinymce_ИмяПлагина extends editor_tinymce_plugin {
protected $buttons = array ('ИмяПлагина');
```

```
protected function update_init_params(array & $params, context $context,
array $options = null) {
```

```
    $this->add_button_after($params,3,'ИмяПлагина','spellchecker');
```

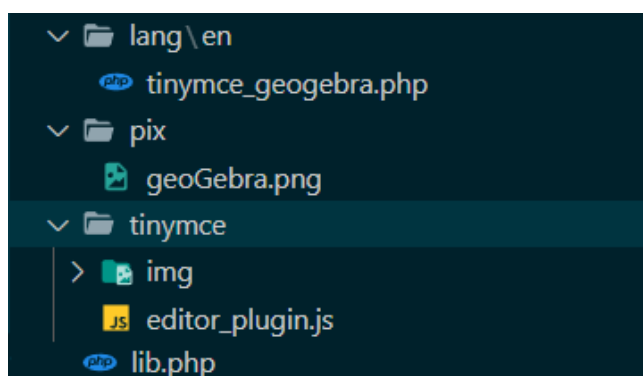
```
    $this->add_js_plugin($params);
```

```
    }
}
```

version.php содержит информацию о версии плагина Moodle

```
<?php
defined('MOODLE_INTERNAL') || (die);
#текущая версия плагина (Дата: ГГГГММДДЧЧ)
$plugin -> version = 2012112900;
#требующиеся версия Moodle (Дата: ГГГГММДДЧЧ)
$plugin -> requires = 2012112900;
#полное имя плагина
$plugin -> component = 'tinymce_ИмяПлагина';
```

В итоге мы должны получить примерно такой каталог:



ГЛАВА 2

Однако, глава

2.1 и раз

2.1.1 и два

и три

Рисунок 2.1 – sdfgsdfgsf

Таблица 2.1 – sdfgsdfgsf

a	b
a	b
a	b

Заключение

Библиографический список

1. Библиография оформляется по ГОСТ 7.0.5-2008
2. Библиография оформляется по ГОСТ 7.0.5-2008

ПРИЛОЖЕНИЕ А

Листинг программы

```
1 f(x, F) := block([i, S],
2   S: zeromatrix(dim, dim),
3   for i: 1 thru length(F) do
4     S: S+mod(F[i]*(x^(i-1)), P),
5   return(mod(S, P))
6 );
```