

*Зайцев Константин Александрович, магистрант,*

*Московский Автомобильно-дорожный Институт – МАДИ, Россия, г. Москва*

## МЕТОДОЛОГИЯ РЕАЛИЗАЦИИ МОДУЛЕЙ В СДО MOODLE

**Аннотация:** В данной статье рассмотрено понятие «системы дистанционного обучения» на примере СДО Moodle. Исследована и описана методология реализации модулей в СДО Moodle.

**Ключевые слова:** дистанционное обучение, электронное обучение, СДО, Moodle, плагины, модули, php.

**Annotation:** This article discusses the concept of « learning management systems» on the example of LMS Moodle. The methodology for implementing modules in Moodle is studied and described.

**Keywords:** distance learning, e-learning, LMS, Moodle, plugins, modules, php.

Системы дистанционного обучения (СДО или LMS, от английского – learning management systems) позволяют организовать учебный процесс «с нуля» и отслеживать успешность учеников посредством создания онлайн-курсов или виртуальных классов, доступных в любое время и в любой точке мира, где есть интернет [1]. Все учебные материалы при этом хранятся в одном месте, их удобно адаптировать и пересматривать в зависимости от целей обучения и сферы деятельности компании или организации.

Moodle – система дистанционного обучения предназначенная для предоставления всем участникам учебного процесса единой надежной, безопасной и интегрированной системы для создания персонализированной учебной среды [2]. Является полностью бесплатной, распространяется под

лицензией General Public License и обеспечивает высокий уровень функциональности в создании и управлении образовательными онлайн-курсами.

Основными преимуществами данной системы является модульность, масштабируемость и открытость исходного кода. Moodle позволяет создавать базовые модули и добавлять элементы с разной степенью сложности в программу.

Процесс реализации состоит из 4-х основных этапов, характерных для всех типов модулей для СДО Moodle:

- подготовка;
- разработка;
- тестирование;
- установка.

Данные этапы не представляют собой непрерывный последовательный процесс. После этапа тестирования в большинстве случаев необходимо вернуться на этап разработки, поэтому этапы в процессе разработки могут повторяться. Процесс представлен на рис. 1.

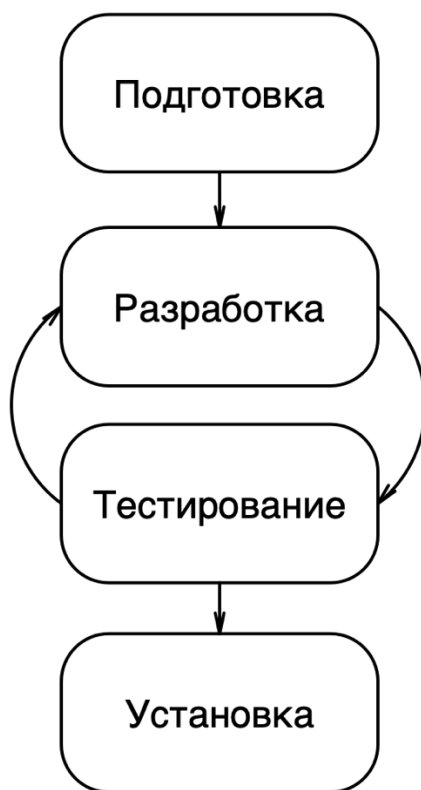


Рисунок 1. Этапы реализации модуля для СДО Moodle

К тому же этап тестирования является необязательным и его вполне можно пропустить, но в данном случае в процессе работы могут возникнуть непредвиденные проблемы.

### **Этап подготовки**

Все установленные и разработанные модули в системе Moodle расположены на веб-сервере в корневом каталоге с исходным кодом в директории mod.

Для разработки нового модуля необходимо создать в директории mod каталог, название которого должно совпадать в наименованием разрабатываемого плагина. Все необходимые файлы, библиотеки и исходные файлы, принадлежащие плагину, должны находится в данном каталоге.

Основопологающим этапом при создании плагина для СДО Moodle, является настройка связи и взаимодействия с базой данных. Для этого в корневом каталоге плагина необходимо создать директорию db. В дальнейшем на этапе разработки, в него будут добавлены файлы необходимые для взаимодействия с базой данных.

Для хранения изображений необходимо создать директорию pix. В данной директории необходимо разместить все графические изображения, которые будут использоваться в плагине. Файлу с иконкой плагина должно быть присвоено наименование – icon.png.

Вспомогательные классы, создаваемые разработчиками плагина, можно хранить в директории classes, позволяющая легко сориентироваться в проекте. Данная директория не является обязательной для работы плагина.

Необязательной является и директория tests. В ней принято располагать все файлы unit-тестирования, реализованных с помощью фреймворка PHPUnit.

Директория backup служит для хранения файлов, предоставляющих возможности резервного копирования и восстановления данных. Данная директория не является обязательной, если плагин не должен предоставлять такие возможности.

Для поддержки мультиязычности и возможности перевода плагина на несколько языков, необходимо создать директорию lang, в которой необходимо разместить соответствующие языковые подпапки. Внутри каждой такой папки должен располагаться файл с расширением php и названием, соответствующим наименованию плагина. Например, название папки en соответствует английскому языку, а ru – русскому. Файл с языковыми настройками состоит из ассоциативного массива \$string, ключом которого является кодовое название строки, а значением – строка на нужном языке. Благодаря встроенной функции get\_string () в любом месте исходного кода Moodle можно получить необходимую строку. Для этого данной функции необходимо передать два параметра – название плагина и кодовое название строки.

Наименование, версию плагина, метаданные, минимальные системные требования к системе Moodle и дополнительные используемые плагины необходимо описать в файле version.php расположенном в корневом каталоге разрабатываемого модуля.

Обобщенная файловая структура для разработки плагина в СДО Moodle представлена на рис. 2.



Рисунок 2. Обобщенная файловая структура плагина в СДО Moodle

### Этап разработки

Этап разработки состоит из создания и написания исходных файлов, отвечающих за функциональность плагина, которые располагаются в описанных выше директориях [3].

Первоначально необходимо сформировать файл `install.xml` в каталоге `db` корневой директории плагина. В данном файле в формате `xml` необходимо описать таблицы, связи между ними и другими существующими таблицами. При первой установке плагина будут созданы все таблицы и связи, описанные в данном файле.

Благодаря встроенному редактору для файлов `install.xml`, который доступен главному администратору системы после установки СДО Moodle, имеется возможность создавать, просматривать и изменять файлы установки, а также просматривать связи между таблицами.

Файл `update.php`, расположенный в каталоге `db` служит для внесения изменений в существующие таблицы плагина или добавления новых таблиц.

В зависимости от версии это позволяет быстро модифицировать базу данных.

Файл `index.php` содержит в себе визуальное представление всех экземпляров данного элемента курса.

Файл `mod_form.php` служит для отображения веб-интерфейса плагина в режиме редактирования.

Файл `access.php`, расположенный в корневом каталоге плагина, содержит права пользователей системы в зависимости от их роли и уровня доступа. Данный файл носит обязательный характер.

Файл `view.php` содержит веб-интерфейс отображения плагина в режиме использования на странице курса.

Основным файлом, в котором реализуются все основные функции и методы необходимые для работы разрабатываемого плагина является – `lib.php`. При определении собственных функций в данном файле необходимо придерживаться стилистике наименований используемой в Moodle. Названия функций должны начинаться с наименования плагина.

Файл `settings.php` необходим, если плагин должен иметь функционал настройки со стороны администратора системы. Например, в настройках плагина «Тест» предоставляется возможность указать ограничение по времени, задать количество попыток, максимальную оценку и метод оценивания.

### **Этап тестирования**

Этап тестирования необходим для проверки разработанных на предыдущем этапе функций, а также написания тестов для ещё не реализованных функций. Написание тестов в СДО Moodle происходит с использованием популярного фреймворка PHPUnit, основанного на модульном тестировании.

Функциональные возможности PHPUnit [4]:

- тестирование исключений;
- провайдеры данных;
- создание модульных тестов;

- тестирования кода, взаимодействующего с базой данных;
- поддержка фиктивных объектов;
- генераторы отчетов;
- интерфейс командной строки для выполнения тестов.

Для тестирования функций, реализованных в файле `lib.php` необходимо в директории `tests` создать файл `lib_test.php` и унаследовать класс от `advanced_testcase`.

Перед запуском каждого теста вызывается метод `setUp`, после выполнения теста вызывается метод `tearDown`. Методы `setUpBeforeClass` и `tearDownAfterClass` вызываются единожды перед началом и окончанием всех тестов соответственно. При запуске тестирования фреймворк PHPUnit начинает выполнение всех методов, начинающихся со слова `test` [5].

Внутри тестовых методов используется один из методов проверки, начинающихся с префикса `assert`. К примеру, при использовании метода `assertEquals` тест считается успешным, если плановое и фактическое значения равны. Пример тестового класса представлен на рис. 3.

```

1  <?php
2
3  class mod_ytstream_testcase extends advanced_testcase
4  {
5
6      public static function setUpBeforeClass()
7      {
8          global $CFG;
9          require_once($CFG->dirroot . '/mod/lib.php');
10     }
11
12     public function test_deleting()
13     {
14         $this->resetAfterTest(true);
15         $DB->delete_records('user');
16         $this->assertEmpty($DB->get_records('user'));
17     }
18
19     public function test_user_table_was_reset()
20     {
21         $this->assertEquals(2, $DB->count_records('user', array()));
22     }
23 }

```

Рисунок 3. Пример тестового класса

Перед выполнением тестов в файле `config.php`, расположенном в корневой директории Moodle необходимо сконфигурировать тестовую среду, а именно указать сервер базы данных файловый сервер для тестов и произвести её инициализацию командой `php admin/tool/phpunit/cli/init.php` [6].

Тестирование происходит через консольную утилиту `phpunit`. При запуске утилиты без параметров, происходит тестирование всех плагинов в системе. Чтобы протестировать определенный файл, утилите необходимо указать путь к файлу и название тестируемого класса. Запуск тестов с помощью консольной утилиты приведен на рис. 4.



```
→ moodle vendor/bin/phpunit --debug mod_bookref_lib_testcase mod/bookref/tests/lib_test.php
Moodle 3.1.4+ (Build: 20170202)
Php: 7.0.15.0.0.16.04.4, mysqli: 5.7.17-0ubuntu0.16.04.2, OS: Linux 4.4.0-72-generic x86_64
PHPUnit 4.8.27 by Sebastian Bergmann and contributors.

Starting test 'mod_bookref_lib_testcase::test_bookref_type_book_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_article_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_dissertation_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_book_conference_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_abstract_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_preprint_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_report_name_is_valid'.
Starting test 'mod_bookref_lib_testcase::test_bookref_type_internet_name_is_valid'.
Time: 393 ms, Memory: 32.00MB
OK (8 tests, 8 assertions)
```

Рисунок 4. Запуск тестов с помощью консольной утилиты

Помимо консольной утилиты, запускать тесты позволяют большинство сред разработки поддерживающих PHP, такие как NetBeans, PHPStorm, и Eclipse.

### Этап установки


Установить разработанный плагин типа «элемент курса» в СДО Moodle можно двумя способами.

Первый способ заключается в том, что администратору в настройках системы необходимо перейти в раздел «Плагины», затем перейти к пункту «Установка плагина». Далее необходимо загрузить архив в формате zip с исходными файлами плагина [7].

Второй способ заключается в копировании исходных файлов плагина в каталог mod, расположенный в корневой директории Moodle на веб-сервере. Следующим шагом необходимо перейти на сайт. Администратору системы будет показано предупреждение о плагинах, требующих внимания [8].

## Плагины, требующие внимания

Плагины, требующие внимания 2 Все плагины 406

Название плагина / Каталог	Текущая версия	Новая версия	Необходимо	Источник / Статус
<b>Модули элементов курса</b>				
 YouTube Stream <small>/mod/ytstream</small>	2020031106	2020031107	• Moodle 2017051500	<a href="#">Дополнение</a> <a href="#">Должен быть обновлен</a>

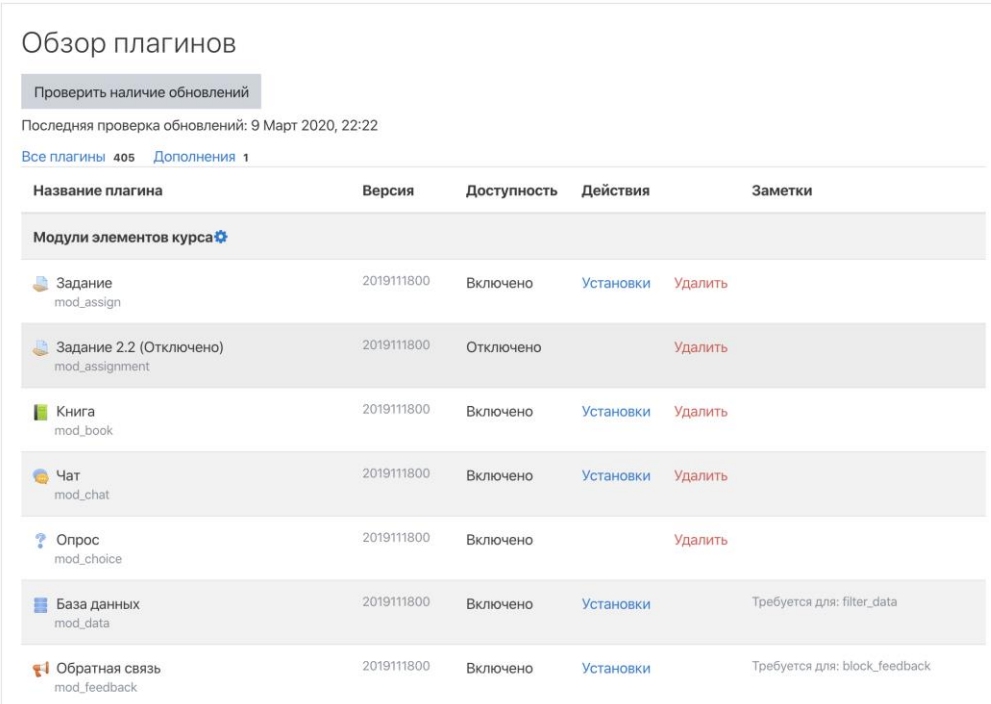
[Обновить](#)

[Обновить Moodle](#)

Рисунок 5. Уведомление о плагинах, требующих внимания

Во время установки система Moodle обновляет и создает необходимые таблицы в базе данных для корректной работы плагинов.

В пункте меню «Обзор плагинов» раздела «Плагины» отображаются все установленные в системе плагины (рис. 5.). В данном разделе можно увидеть версию плагина, изменить его настройки (если предусмотрена такая возможность), а также удалить плагин [9].










Обзор плагинов				
<a href="#">Проверить наличие обновлений</a>				
Последняя проверка обновлений: 9 Март 2020, 22:22				
<a href="#">Все плагины</a> 405 <a href="#">Дополнения</a> 1				
Название плагина	Версия	Доступность	Действия	Заметки
<b>Модули элементов курса</b>				
 Задание <small>mod_assign</small>	2019111800	Включено	<a href="#">Установки</a> <a href="#">Удалить</a>	
 Задание 2.2 (Отключено) <small>mod_assignment</small>	2019111800	Отключено	<a href="#">Удалить</a>	
 Книга <small>mod_book</small>	2019111800	Включено	<a href="#">Установки</a> <a href="#">Удалить</a>	
 Чат <small>mod_chat</small>	2019111800	Включено	<a href="#">Установки</a> <a href="#">Удалить</a>	
 Опрос <small>mod_choice</small>	2019111800	Включено	<a href="#">Удалить</a>	
 База данных <small>mod_data</small>	2019111800	Включено	<a href="#">Установки</a>	Требуется для: filter_data
 Обратная связь <small>mod_feedback</small>	2019111800	Включено	<a href="#">Установки</a>	Требуется для: block_feedback

Рисунок 6. Страница со списком установленных плагинов

Для добавления плагина в курс необходимо перейти в «Режим редактирования» на странице курса, затем нажать на кнопку «Добавить элемент

или ресурс» и выбрать необходимый элемент курса. Пример окна с добавлением элемента или ресурса курса представлен на рис. 7.

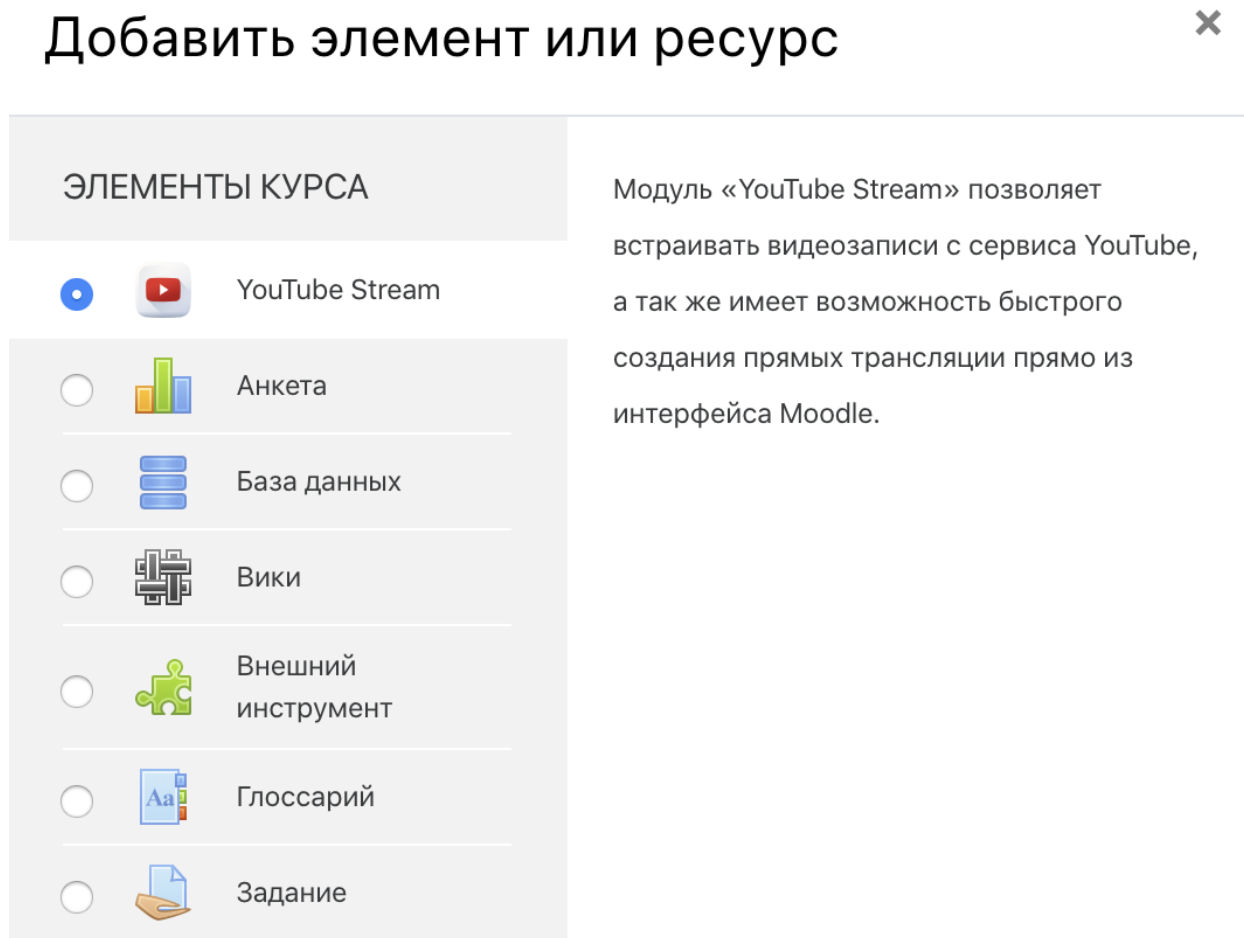


Рисунок 7. Окно добавления элемента курса

### **Библиографический список:**

1. Системы дистанционного обучения. [Электронный ресурс] URL: <https://evergreens.com.ua/ru/articles/best-lms-2020.html> (дата обращения: 15.03.2020).
2. Moodle. Open-source learning platform. [Электронный ресурс] URL: <https://moodle.org> (дата обращения: 15.03.2020).
3. Activity modules. [Электронный ресурс] URL: [https://docs.moodle.org/dev/Activity\\_modules](https://docs.moodle.org/dev/Activity_modules) (дата обращения: 18.03.2020).
4. PHPUnit. [Электронный ресурс] URL: <https://phpunit.de> (дата обращения: 03.04.2020).

5. API. [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/API> (дата обращения: 28.03.2020).
6. Steve Foreman The LMS Guidebook: Learning Management Systems Demystified. — Association for Talent Development, 2017. — 248 с.
7. Jason Cole, Helen Foster Using Moodle: Teaching with the Popular Open Source Course Management System. — O'Reilly Media, 2007. — 284 с.
8. Ruth C. Clark, Richard E. Mayer e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning. — Wile, 2016. — 528 с.
9. Steve Foreman The LMS Guidebook: Learning Management Systems Demystified. — Association for Talent Development, 2017. — 248 с.