

## Экзамен

# №1 Интеграция дизайна и настройка платформы

по платформе «Bitrix Framework»

---

«Академия 1С-Битрикс», 2017 г.

Версия 2.0.15, 28.12.2017 г.

Документ обновляется, актуальная версия доступна по ссылке <https://academy.1c-bitrix.ru/~ex1desc>

## Входные требования

- Наличие базовых знаний HTML, CSS, JS
- Знание основ программирования на PHP
- Сертификат по онлайн-тесту: «Администратор. Базовый»

[http://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=35](http://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=35)

## Предмет экзамена

Оцениваются навыки по реализации проектов в рамках типового функционала: установка системы и настройка, интеграция HTML верстки, создание структуры сайта, использование компонентов, информационных блоков, поиска и ряда других возможностей.

## Материалы для подготовки

- Курс "№1 - Интеграция дизайна и настройка платформы"

<http://academy.1c-bitrix.ru/training/course/5862/>

- Курс Разработчик Bitrix Framework

[http://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=43](http://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43)

- Документация

[http://dev.1c-bitrix.ru/api\\_help/](http://dev.1c-bitrix.ru/api_help/)

## Формат экзамена

- Длительность: 4 часа
- Решение практических заданий путем настройки Bitrix Framework, интеграции готовой HTML верстки и непосредственного программирования на языке PHP с использованием API Bitrix Framework.
- Удалённое подключение через терминал к экзаменационному серверу

<https://academy.1c-bitrix.ru/certification/exam-server.php>.

**Внимание!** При подготовке учитывайте, работа с сервером по терминалу - не такая быстрая, как локальная разработка. Тестовые билеты нужно решать с запасом времени. Познакомьтесь с окружением заранее, чтобы во время экзамена не отвлекаться на незнакомую среду.

- Во время экзамена можно пользоваться только справочными материалами, доступными на экзаменационном сервере: локальные СНМ файлы с документацией по API Framework и PHP. «Гуглить» и использовать «свои заготовки» - запрещено.

## Оценка решения

Для успешного прохождения экзамена необходимо выполнение нескольких критериев:

- **Все задания из билета должны быть решены**

Объем заданий рассчитан с учетом работы по терминалу, временем на самопроверку и запасом на возможные перепоключения к серверу.

**Внимание!** Методика проведения экзамена подразумевает, что кандидат знает необходимые технологии и имеет практику решения задач, подобных экзаменационным. Разбираться во время экзамена как решается задача - не будет времени. Конкретные API конечно не нужно помнить, они доступны в локальном файле документации.

- **Правильно используются технологии Bitrix Framework**

Экзамен – это оценка навыков правильного использования технологий Bitrix Framework. К темам указаны правила решения заданий. При несоблюдении правил экзамен не будет успешно пройден.

**Внимание!** При нарушении хотя бы одного правила выделенного знаком (!) – экзамен не будет засчитан, даже если все остальное верно.

- **Решение должно работать верно**

Для зачета решения задания необходимо чтобы по факту все работало верно. Если задание решено с учетом технологий, но, например, из-за опечатки не работает верно, то решение не будет засчитано.

**Внимание!** У кандидата не будет возможности после экзамена объяснить, что он опечатался или он знал, но забыл сделать что-то во время экзамена. Проверяйте, что ваше решение по факту работает верно.

# Темы экзамена

## 1. Установка и первичная настройка системы, административный интерфейс

- Установка 1С-Битрикс, установка обновлений, проверка окружения
- Интерфейс администратора в публичной части, административный раздел сайта.
- Настройки модулей: главный, управление структурой, информационные блоки и другие.

**Обратите внимание.** На экзамене система уже будет установлена, обновлять – не нужно. Это упрощение сделано ввиду решения заданий в рамках ограничения времени и некоторых ограничений экзаменационного сервера.

## 2. Основы архитектуры Bitrix Framework

- Архитектура продукта. Понятие модели MVC и ее реализации в 1С-Битрикс: модуль (API) – шаблоны – компоненты. Порядок выполнения страницы.
- Элементы системы: сайт и его настройки, шаблон сайта, файловая структура публичного раздела, компоненты, комплексные компоненты, шаблоны компонентов, статическая и динамическая информация, языковые файлы, управление доступом.
- Основы мультиязычности, использование LANG файлов.
- Настройка системы, использование файлов /bitrix/php\_interface/dbconn.php, after\_connect.php, /bitrix/.settings.php, .settings\_extra.php
- Использование файла init.php
- Использование папки local для хранения своих объектов.

**Общие правила для решения заданий:**

- **(!)** Ядро продукта не изменяется: типовые компоненты, их типовые шаблоны, модули и т.д. Шаблоны компонентов копируются перед кастомизацией.
- **(!)** Задания решаются штатными компонентами и созданием для них новых шаблонов, а так же настройками и API. Создание собственного компонента в рамках этого экзамена означает незнание возможностей системы, «изобретение велосипеда».
- **(!)** Внедренный шаблон сайта и компонентов выглядят аналогично исходной верстке, нет «поехавших» элементов и «поломанной» верстки. Допускается небольшое смещение верстки только при включенном режиме правки сайта.
- **(!)** Созданный шаблон сайта, шаблоны компонентов должны соответствовать внедряемой верстке. Сохранение каких-либо тегов, их атрибутов из стандартного

шаблона сайта или стандартных шаблонов компонентов – является значительной ошибкой. Внедренный шаблон перестает соответствовать исходной HTML верстке, что приводит к непредсказуемым последствиям в отображении сайта.

- **(!)** Компоненты, выводящие динамическую информацию – кешируют результат.
- При внедрении шаблона сайта, шаблонов компонентов – удаляются файлы которые создаются при копировании типовых шаблонов, но по факту не используются в созданном шаблоне.

### 3. Управление публичной структурой сайта

- Управление страницами и разделами сайта, иерархия разделов.
- Свойства страниц и разделов:
  - Управление стандартными свойствами: заголовок, описание, ключевые слова.
  - Наследование значений свойств.
  - Создание собственных свойств.
- Управление значениями свойств страниц и разделов с помощью отложенных функций
  - Принцип работы отложенных функций
  - Типовые API функции
  - Использование связки функций ShowProperty, SetPageProperty, GetProperty.
- Навигация на сайте: меню и «цепочка» навигации.
- Принцип работы меню: тип, иерархия разделов, многоуровневое меню.
- Принцип работы цепочки навигации.

#### Правила:

- **(!)** Соблюдена физическая иерархия разделов, файлов и подразделов, созданных на сайте. Вложенные разделы и страницы должны не только верно отображаться в меню, но и физически находится в соответствующих разделах.
- **(!)** Отложенные функции применяются верно. Значение, установленное ниже чем его вывод, отображается корректно.
- Используется наследование свойств разделов и страниц сайта «вниз по иерархии», а не прописываются одинаковые значения у страниц и разделов «вручную».
- В меню сайта нет лишних, дублирующих или не соответствующих структуре сайта пунктов.

#### 4. Интеграция HTML шаблона

- Устройство шаблона сайта: расположение в системе, файловая структура шаблона, используемые API.
- Принцип создания шаблона: разделение на header, footer и контентную часть.
- Использование стандартных файлов и разделов шаблона для хранения ресурсов: стилей, скриптов, изображений, и т.д.
- Подключение дополнительных ресурсов шаблона (скриптов, стилей, шрифтов и т.д.).
- Использование возможности объединения и сжатия стилей и скриптов, перенос подключения скриптов вниз страницы.
- Создание нескольких шаблонов сайта, хранение общих ресурсов в шаблоне .default
- Настройка отображения разных шаблонов сайта по условиям.
- Реализация страниц сайта со сложными элементами верстки, не подразумевающая непосредственное изменение контент-менеджером.
- Базовые понятия поддержки мультязычности – использование языковых файлов в шаблоне сайта.

##### Правила:

- **(!)** Должно быть корректное разделение HTML шаблона на header и footer, вызов #WORK\_AREA#. Повторяющиеся элементы вынесены в шаблон. Контент страницы это область только для контент-менеджера, в которой он может полностью все удалить, при этом отображения сайта должно остаться целостным.

Страницы содержащие сложную верстку и не предполагающие изменение контент-менеджером либо получают ограничение доступа, либо отдельный шаблон, не оставляющий в контентной части верстку, изменение которой может «сломать» сайт.

- **(!)** При внедрении шаблона сайта не допускается «бездумный копипаст» типового шаблона и изменение его под новый шаблон. Все теги, начиная с первой строки внедряемого шаблона, <!DOCTYPE>, должны быть взяты только из внедряемого HTML.
- **(!)** После завершения внедрения в консоли браузера нет ошибок 404, все ресурсы подключены и подгружаются без ошибок.
- Из верстки убраны теги, которые будут выведены ядром, например - указание кодировки <meta charset="">
- Используется защита файлов header.php и footer.php от прямого подключения.

- Подключение дополнительных скриптов и стилей произведено с помощью API платформы, а не напрямую в HTML `<link rel="stylesheet">`, `<script type="text/javascript">`
- Выполнено подключение служебной части - ShowHead;
- Подключена административная панель - ShowPanel;
- Использовано API ShowTitle для установки заголовка браузера.
- Использовано API ShowTitle для установки заголовка страницы, который обычно задан в виде тега H1, перед содержимым страницы.
- В header.php и footer.php пути к картинкам и другим ресурсам указаны через константу SITE\_TEMPLATE\_PATH
- Язык шаблона установлен через константу: `<..... lang=<?=LANGUAGE_ID?>>`

## 5. Компоненты

- Типовые компоненты платформы и выбор компонента для решения типовых задач.
- Устройство компонентов: параметры, реализация логики, шаблоны. Разделение логики и «представления».
- Размещение компонентов в шаблоне и на страницах сайта
- Создание шаблонов для компонентов. Данные, доступные в шаблоне.
- Выбор места для хранения шаблона компонента: сам компонент, шаблон сайта, шаблон сайта default.
- Подключение дополнительных ресурсов в шаблоне компонента через API addExternalCss и addExternalJS;

### Правила:

- В шаблонах компонентов нет «HTML – заглушек», вывод данных в шаблоне строится на основании данных переданных компонентом
- Ссылки на объекты (например на детальный просмотр) строятся на основании данных компонента, а не прописываются «в ручную».

## 6. Контент на сайте

- Компонент «Включаемая область», область решаемых задач.
- Работа с контентом в визуальном редакторе
- Создание стилей для оформления контента, доступных в визуальном редакторе.
- Создание шаблонов, используемых при добавлении новых страниц сайта.

- Создание сниппетов

#### Правила:

- Сделан осмысленный выбор типа подключения файла для компонента `bitrix:main.include`. Внедрить редактирование телефона в шапку сайта, указав свойство «Показывать включаемую область» «для страницы» - означает не понимание сути работы этого компонента.
- При необходимости используется рекурсивное подключение файлов для включаемых областей, а не дублируется одинаковый файл по всем разделам.

## 7. Информационные блоки

- Организация хранения динамической информации на сайте
- Какие задачи решают и для чего применяются.
- Как создавать информационные блоки.
- Основные настройки: привязка к сайтам, индексация для поиска, шаблоны ссылок, поля, свойства, доступ, журналирование.
- Настройка отображения списка элементов и редактирования элемента в административном разделе, параметры фильтра.
- Особенности настройки параметров для верного отображения результатов поиска по контенту информационного блока.
- Экспорт и импорт данных информационных блоков.
- Настройка SEO в информационных блоках.

#### Правила:

- **(!)** Правильно настроены шаблоны ссылок для работы поиска.
- Настроены права доступа – публичная информация видна на сайте не авторизованному пользователю.

## 8. Вывод динамической информации на сайте

- Основные типовые компоненты для вывода динамической информации: списков, элемента детально. Как устроены, какие задачи решают, основные настройки.
- Внедрение компонентов в шаблон сайта и на страницы сайта.
- Работа с `arRelult`: понимание доступных данных. Использование циклов и условий для отображение данных компонента в соответствия с внедряемой версткой.
- Построение ссылок на детальный просмотр на основании данных компонента.

- Создание пунктов меню на основе разделов инфоблока.
- Уменьшение изображений «на лету», CFile::ResizeImageGet.

#### Правила:

- Настройки компонента не оказывают ошибочного влияния на сайт. Не включена опция «установить заголовок страницы» или «добавлять пункт в цепочку навигации» в компонентах внедренных в шаблоне сайта, это приведет к отображению не верных данных на всех страницах сайта.
- В шаблонах компонентов нет «HTML – заглушек», все строиться на основании данных переданных компонентом
- Ссылки на подробный просмотр элементов строятся не «вручную», а с помощью данных из результата работа компонента.

## 9. Комплексные компоненты

- Предназначение и устройство комплексного компонента.
- Путь для поиска и порядок подключения шаблона простого компонента в составе комплексного.
- Типовые компоненты для вывода динамической информации: «новости» для плоских списков, «каталог» для иерархических, «фотогалерея».
- Основные настройки типовых комплексных компонентов.
- Создание темы (шаблона) комплексного компонента.
- Настройка ЧПУ, как работает ЧПУ, «многоуровневое ЧПУ», ЧПУ по символьным кодам.
- Режим работы “AJAX” в типовых компонентах.
- Реализация «Эрмитажа», управления информацией над сайтом. Редактирование и создание элементов информационных блоков.
- Реализация постраничной навигации, свои шаблоны постраничной навигации
- Задание метаданных для динамических страниц сайта с помощью настройки компонента и с помощью настроек SEO в инфоблоках.

#### Правила:

- **(!) Комплексный компонент не используется в шаблоне сайта. Это грубая ошибка и не понимание предназначения комплексных компонентов.**
- В шаблонах компонентов нет «HTML – заглушек», все строиться на основании данных переданных компонентом
- Ссылки между элементами: на подробный просмотр, с список и т.д. строятся не «вручную», а с помощью данных из результата работа компонентов.



- Типовые компоненты применяются правильно: для реализации «плоского» списка компонент «новости», для иерархичной структуры – «каталог».
- Для решения задач по заданию метаданных динамических страниц используется соответствующая настройка информационных блоках (SEO) и типовые возможности компонентов, а не «вручную», с помощью API.

## 10. Кеширование в компонентах

- Суть технологии кеширования, как она реализована в «1С-Битрикс: Управление сайтом»
- Кеширование в типовых компонентах, параметры кеширования компонентов.
- Управляемое кеширование: принцип работы, преимущества этого режима.
- Случаи, когда кеширование может не использоваться.
- Кеширование в компонентах и отложенные функции.

### Правила:

- **(!) Во всех компонентах выводящих динамическую информацию кеширование должно быть включено. Существуют задачи, когда кеширование может не использоваться, но они не рассматриваются в рамках этого экзамена.**
- **(!) Все решенные задачи по сайту – работают при включённом кешировании. Не работающее решение или отключенное кеширование для его работоспособности – это грубая ошибка. Например:**
  - Решение задачи установки метаданных страниц должно обязательно работать и при кешировании.
  - «Эрмитаж» работает при кешировании.

- Кеширование в компонентах настраивается исходя из логики решаемой задачи. Управляемый кеш чаще всего будет уместнее, чем регулярный сброс кеша по интервалу времени, если информация меняется не ежеминутно. При выборке меняющейся информации уместнее выставить сброс кеша по интервалу времени.

Речь о типовых проектах, в случае сложной логики и высоких нагрузок – типовые подходы уже могут быть не эффективны, но в рамках данного экзамена такие задачи не рассматриваются.

## 11. Поиск

- Принцип реализации поиска в 1С-Битрикс, построение поискового индекса.
- Основные настройки работы поиска на сайте.

- Возможности типовых компонентов для реализации поиска.
- Настройка индексации в инфоблоках.
- Построение ссылок на динамическую информацию в результатах поиска.
- Ограничение области поиска.

**Правила:**

- **(!) Ссылки на динамические элементы строятся верно, переход осуществляется на найденный элемент**
- Индексируемые элементы (статические страницы, новости, отзывы, элементы каталога) находятся поиском.

## **12. Почтовая система**

- Почтовые события и почтовые шаблоны
- Типовые почтовые события и шаблоны.
- Почтовые уведомления по пользователем: регистрация, подтверждение регистрации, изменение пользователя, смена и восстановление пароля.
- Создание своих почтовых шаблонов и типов почтовых событий.

## **13. Управление доступом и безопасность**

- Управление доступом к страницам и разделам сайта.
  - Ограничение доступа к разделам и страницам сайта в публичной части сайта.
  - Доступ на редактирование к разделам и страницам сайта в публичной части сайта.
- Отображение страниц и разделов сайта в меню в соответствии с правами доступа.
- Управление доступом к модулям в административной части.
- Создание дополнительного «администратора» сайта с ограниченными правами на: редактирование разделов и страниц в публичной части сайта, изменений данных в информационных блоках, управления модулями в административном разделе.
- Типовые компоненты для регистрации, авторизации, редактирования профиля.
- Кастомизация внешнего вида компонентов авторизации.
- Реализация типовых сценариев: авторизация, выход, регистрация, восстановление пароля, просмотр и редактирование профиля.
- Возможности авторизации: авторизация через соцсети, использование captcha.