

Битрикс экзамен Интеграция дизайна Инфоблоки

Общие практики

- Настраивать журнал изменений и удалений, если просят
 - Настраивать названия (*Новости* → *Сотрудники* или *События* и.т.д)
-

ЧПУ

- Отметить галочку поддержки ЧПУ в компоненте на странице
 - Настроить автогенерацию символьных кодов с переводом
 - Каталог ЧПУ не трогать
 - Создать в инфоблоке для элементов *символьные коды* вручную
 - `#ELEMENT_ID#` → `#ELEMENT_CODE#` и больше ничего не трогать (для поиска — настроить путь в инфоблоке)
-

Управление динамической информацией на сайте

- Создать свой тип инфоблока *Отзывы* или другой
- При импорте XML выбрать этот тип инфоблока
- Включить автогенерацию символьных кодов
- Настроить поля в списке элементов
- В фильтре проверить наличие поиска по определённому полю
- Настроить форму редактирования как просят через кнопку коги

- Все остальные используемые поля — имеются в виду буквально все остальные поля
 - Изменение элементов должно фиксироваться = *включить журнал изменений*
-

Компоненты

Раздел с динамической информацией на сайте

- Поместить на страницу раздела комплексный компонент *Новости*
- Отключить в компоненте установку заголовков, стилей, включение в цепочку навигации
- Выбрать отображаемые свойства на общей и детальной странице в настройках
- Указать название категорий *Новости* → *Отзывы*, *Сотрудники* и.т.д
- Редактировать
local/components/bitrix/news/*template_name*/news.list/.default/template.php
— страницу со списком новостей
- Редактировать
local/components/bitrix/news/*template_name*/news.detail/.default/template.php
— детальную страницу
- Редактировать кнопку "Назад к списку новостей" → "К списку отзывов" в файле **detail.php**
- Редактировать
local/components/bitrix/news/*template_name*/detail.php
— изменить фразу ***вернуться к списку элементов*** с помощью ***lang* файлов**
Назад к списку* → *К списку отзывов
- В поле *Документы* указать возможность загрузки только *pdf* файлов

- Для того, чтобы вытащить название документа, использовать:

```
foreach($arResult["PROPERTIES"]["FILE"]["VALUE"] as $arFile):  
    $filePath = CFILE::GetPath($arFile); $fileOb =  
    CFILE::GetByID($arFile); $file = $fileOb->Fetch();
```

Где название файла — `$file["ORIGINAL_NAME"]`

Ссылка на скачивание — `<a href="<?$filePath?>" download><?$file["ORIGINAL_NAME"]?>`

- Установить название соответствующее контенту для пагинации в настройках компонента
- Проверить установлена ли картинка для анонса: `if (is_array($arItem["PREVIEW_PICTURE"]))`
Сама картинка доступна через `$arItem["PREVIEW_PICTURE"]`
`["SRC"]`

Комплексный компонент сотрудников

- Настройка отображения картинки:

```
if (isset($arItem["PREVIEW_PICTURE"]["SRC"])) { $resizedFile =  
CFile::ResizeImageGet($arItem["PREVIEW_PICTURE"]["ID"], array(  
'width' => ширина, 'height' => высота ),  
BX_RESIZE_IMAGE_PROPORTIONAL, true); $img = $resizedFile["src"]};  
}
```

Цепочка навигации

- Не включать компоненты в цепочку навигации, проверять правильность заголовков
- Не включать / включать страницу в цепочку навигации
- Проверять мешает ли инфоблок/компонент цепочке навигации и исправлять в настройках

Компонент авторизации и регистрации

- Ссылка для кнопки разлогина:
`<a href="?logout=yes&sessid=<?=$_SESSION["fixed_session_id"];?>">`
 - Соц сети и капчу своровать со стандартного компонента
 - Ссылки в экзамене на разлогин, забытый пароль и остальное вводить так:
`/login/?forgotpassword=yes;` Остальные вводятся аналогично
 - Логин пользователя: `$USER->GetLogin();`
 - Имя пользователя: `$USER->GetFullName();`
 - Ссылки на регистрацию, забытый пароль и.т.д — в коде вызова компонента. `$arParams...`
-

Поиск

- Установить галочки для свойств инфоблока *Участвуют в поиске*
 - Обязательно выполнить переиндексацию (снять галочку *переиндексировать только изменённые*)
 - Проверить работает ли поиск по свойству *Должность* и *Компания* или других по заданию
-

Перевод шаблона в битрикс

- Использовать `$APPLICATION->GetCurDir()=="/"` если просят всё сделать одним шаблоном для добавления элементов только на главной странице или наоборот
- Закинуть **favicon.ico** (исправить путь в `<link rel=""...>`)
- Заголовок страницы должен быть в *header'e*
- Все добавления CSS → `SetAdditionalCSS(path);`
- Все добавления JS → `AddHeadScript(path);`
- Установка *метадааты* (*метатеги* для кодировки, ключевых слов и описания) не нужна, потому что `ShowHead();`
- Установка заголовка страницы — `ShowTitle();`

- Вывод заголовка страницы — `ShowTitle(false);`
- Установка языка `<html lang="<?=LANGUAGE_ID;?>">`
- Константы `SITE_DIR`, `SITE_TEMPLATE_PATH` для картинок и методов
- Обязательно вставить кусок кода ниже в начало *footer* и *header*

Метаданные для разделов и страниц

- В настройках *инфоблока* метаданные для элемента указать как просят
- В файлах и папках можно задать метаданные для раздела (папки)
- Для конкретной страницы если нужно переопределить *метадату* (дропдаун меню изменения страницы)

Внедрение в шаблон компонентов

- Создать компонент на странице + копировать шаблон
- Изменить код `template.php` и подогнать верстку шаблона под нужную
- Вырезать код вызова шаблона и записать в *header* или *footer*
- Для простых компонентов редактируется `template.php`, для комплексных `template.php` конкретных простых компонентов в составе сложных

Включаемые области

- Задание со временем: `$currentHour = date("H", time())` и дальше условие

[ex1-11] Управление текстом в шаблоне блок

Полезная информация

- Найти вёрстку на странице и заменить её на вызов *включаемой области* для раздела
- Скопировать шаблон компонента *включаемой области*
- Интегрировать в шаблон вёрстку в файле `local/templates/exam1/components/bitrix/main.include/"название нового шаблона"/template.php`
- Проверить наличие в вёрстке строки `if ($arResult["FILE"] <> '')`
- Строку `<?include($arResult["FILE"])?>` поместить внутрь вёрстки туда, где должен быть текст
- `AREA_FILE_SHOW ⇒ sect, AREA_FILE_SUFFIX ⇒ inc, AREA_FILE_RECURSIVE ⇒ "Y", EDIT_TEMPLATE ⇒ "", PATH ⇒ ""`
- Для главной страницы добавить *включаемую область* с текстом, в другой раздел — свою
На месте *включаемой области* должен быть эрмитаж, с помощью которого это делается

Шаблоны для статичной страницы (пассивный шаблон)

- Создать папку `page_templates` в папке шаблона
 - Создать внутри `inner.php`, записать туда демо контент и добавить вызов `header` и `footer` (можно скопировать из индексной страницы сайта `/index.php`)
 - Рядом с `inner.php` создать `content.php` и написать там помимо `B_PROLOG_INCLUDED:`
`$TEMPLATE["inner.php"] = array("name" ⇒ "Новый шаблон", sort ⇒ 1);`
-

Почтовый шаблон

- Редактирование события *USER_INFO*
 - Создать почтовый шаблон через *административную панель*
 - Деактивировать старый, если это необходимо (регистрационная информация)
 - Проверить ссылки в почтовом шаблоне, если они имеются
 - Настроить нужный компонент на работу с новым почтовым шаблоном
(например, форма обратной связи)
 - Проверить соответствие ссылок названиям разделам
-

Меню

Создать новый тип меню

- *Настройки главного модуля → Управление структурой → внизу типы меню*
 - Создать свой и назвать как просят
 - Выбрать *тип меню* у компонента, если просят
 - На главной странице создать меню в текущем разделе и добавить ссылки (рекурсия)
 - Натянуть вёрстку на новый шаблон компонента
-

Поддержка стилей отображения пунктов меню

- Открыть вёрстку из материалов к заданию + скопировать стили в **template_styles.css**
- В параметрах меню добавить новый параметр с названием как просят + указать значение
- В шаблоне добавить проверку на наличие этого класса `if (isset($arResult["STYLE"]))` и дописать элементу этот класс (по-

умолчанию установить `$arClass='';` и просто дописать в конец `class='... <?=$arClass?>'` через пробел)

Вывод текста для выпадающего меню

- Создать в меню свойство *DESCRIPTION* и указать для нужных пунктов значение
 - В вёрстке скопировать из примера блок и выводить его, если `isset($arResult["PARAMS"]["DESCRIPTION"])`
-

Создать пользователя с ограниченным доступом

Редактор отзывов

- Создать новую *группу пользователей* с нужным названием
 - Установить *управление структурой* на редактирование файлов и папок
 - В самой *структуре* в свойствах папки **/bitrix/admin** дать доступ на чтение
 - В настройках инфоблока отзывов установить *доступ на изменение*
 - *Зарегистрировать* пользователя с новой группой и логином+паролем от задания
 - В настройках главного модуля установить для этой группы *Показывать панель управления*
-

Администратор опросов

- Создать нового пользователя с логином и паролем как в задании
- Добавить *новый раздел* опросов

- Создать группу *Менеджер опросов* и привязать к ней нового пользователя
 - Дать доступ к опросам и голосованиям на запись
 - Дать доступ к редактированию файлов и папок в управлении структурой
 - В самой структуре в свойствах папки */bitrix/admin* дать доступ на чтение
 - Для раздела опросов дать доступ на запись
 - В настройках главного модуля установить для этой группы *Показывать панель управления*
 - Создать группу опросов и сам опрос для проверки
-

Остальные простые задания

Ускорение работы с контентом HTML

- Включить редактирование страницы и нажать на кнопку слева *Добавить сниппет*
 - Написать название *сниппета* и вставить его содержимое
-

Ланговые файлы

- Все статичные фразы заменять на *lang файлы*, если в задании не указано иначе
-

Ограничение доступа к пункту меню

- В настройках меню добавить условие *Выражение PHP* `$USER->isAuthorized()`
-

Кэширование

- Установить для всех меню авто+управляемое кэширование в конце!
-

Все нужные методы и их сигнатуры

- `CFILE::ResizeImageGet($img, array('width' => w, 'height' => h), BX_RESIZE_IMAGE_PROPORTIONAL)` — изменение раздела картинки
- `CFILE::GetPath($arItem)` — получение пути к документу
- `CFILE::GetById($arItem)` — получение объекта документа
- `$APPLICATION->ShowTitle()` — вывести заголовок страницы
- `$APPLICATION->ShowHead()` — загрузить CSS, JS, и метадату
- `$APPLICATION->SetAdditionalCSS()` — добавить CSS файл
- `$APPLICATION->AddHeadScript()` — добавить JS файл
- `var_dump($var)` — посмотреть содержимое `$var`
- `$APPLICATION->ShowPanel()` — отобразить административную панель (функция должна быть вызвана внутри `<div id='panel'>`
`</div>`)
- `GetMessage('MESSAGE_NAME')` — вывести ланговый файл фразы `MESSAGE_NAME`
- `date("H")` — получить текущее время в часах
- `$APPLICATION->GetCurDir()` — получить текущую директорию, / — главная страница
- `$APPLICATION->ShowProperty("prop_name")` — вывести значение выбранного свойства

Константы и их применение

- `SITE_DIR` — корневая директория сайта
- `SITE_TEMPLATE_PATH` — путь к папке текущего активного шаблона
- `LANGUAGE_ID` — язык, устанавливаемый через битрикс

Поля и свойства разных

`$arResult` и `$arResult`

В комплексных компонентах всегда `$arResult`, а в простых — `$arResult`

- `$arResult["ORIGINAL_FILE_NAME"]` — имя документа

[ex1-25] Создание свойства страницы управление цветом шапки левого меню

- В управлении структурой добавить новое свойство для левого меню
- В файлах и папках найти кнопку свойства и указать значения для всего сайта/раздела
- В шаблоне компонента меню вывести в качестве класса новое свойство

Возможные ошибки

Ошибка: Не получается натянуть вёрстку на шаблон

Подсказка: Опирайтесь нужно на то, что в `#WORK_AREA#` вряд ли будут какие-то *html* тэги кроме `<p>`, ``, ``, `<hr>`. При пустой рабочей области сайт должен выглядеть нормально, в нём не должны отсутствовать большие куски типа меню, каких-то компонентов и.т.д.

Ошибка: Дублируются все пункты меню в подпунктах первого

Решение: Для 1-го пункта указать ссылку не на раздел, `/company/`, а

страницу `/company/index.php`. Либо указать без последнего слеша: `/company`.

Ошибка: *форма авторизации* в шапке не выводит сообщения о неверном логине/пароле.

Решение: Во-первых, в настройках компонента включить опцию *Показывать ошибки*. Во-вторых, в шаблоне компонента нужно сохранить (не удалить при доработке стандартного шаблона) следующие строки:

```
if ($arResult['SHOW_ERRORS'] == 'Y' && $arResult['ERROR'])  
ShowMessage($arResult['ERROR_MESSAGE']);
```

Ошибка: Нет необходимости выводить заголовок в шаблоне компонента

Решение: Заголовок детальной страницы выводится в шаблоне сайта, следует им управлять согласно заданию посредством настройки *SEO* информации в инфоблоках.

Не использовать `#CODE#`, потому что это ломает *ЧПУ* из-за проблем с обратной совместимостью.

Ошибка: Компонент не обновляется

Решение: *Сбросить кэш*.

Ошибка: Вместо свойств и полей в компоненте выводятся пустые строки

Решение: Удостовериться в том, что во всех вызовах используется

```
<?=GetMessage('id')?>
```

. Из-за забытого знака *равно* всё может не работать

Ошибка: Через поиск невозможно попасть на страницу

Решение: Проверить корректность путей в инфоблоке и в настройках компонента поиска.

Ошибка: Поиск по нужным свойствам или полям не работает

Решение: Проверить установленные настройки для свойств и полей *участвует в поиске* и выполнить переиндексацию.

Ошибка: Не отображается пагинация и/или эрмитаж у компонентов

Решение: Для того, чтобы работала пагинация, в шаблоне компонента должны быть условные блоки с текстом

`DISPLAY_TOP_PAGER` и `DISPLAY_BOTTOM_PAGER`.

Для того, чтобы работал эрмитаж, нужно скопировать из существующего шаблона строки, содержащие `$this->AddEditAction...` и `$this->AddDeleteAction...`, затем в `id` блока, который должен использовать эрмитаж, перенести строку `$this->GetEditAreaId...`.

Ошибка: Поехала вёрстка

Решение: Проверить *html* с помощью браузера, если всё выглядит идентично, то проблема может быть буквально в одном лишнем или одном недостающем символе, который был случайно стёрт в процессе кастомизации.

Для экономии времени

Перед экзаменом выучить названия и сигнатуры всех методов для того, чтобы не тратить время на их поиск и эксперименты с ними.

Перед экзаменом выучить все пути от `$arResult`, `$arParams` и `$arResult` до нужных свойств, параметров и полей.

В самом начале экзамена перенести все файлы вёрстки в **www** для экономии времени на поиске и перемещении файлов.

В самом начале экзамена открыть вкладки в *административной панели: инфоблоки, настройки главного модуля, структура сайтов*.

В самом начале экзамена открыть вкладки в *административной панели: инфоблоки, настройки главного модуля, структура сайтов*. Держать эти вкладки всегда открытыми для быстрого доступа к ним.

Сразу после создания своего шаблона перенести в него все компоненты, стили, скрипты и другие файлы.

Сразу после импорта инфоблока настроить всё, связанное с ним, — доступ, участие полей и свойств в поиске, участие полей и свойств в фильтре, генерацию символьных кодов, обязательность параметров и.т.д.

Сразу после добавления компонентов со свойствами на страницу в настройках компонента выбрать эти свойства на отображение.

Во время экзамена сначала полностью прочитать все пункты задания, и только потом начинать их выполнять — некоторые пункты заданий могут выполняться рядом в административной панели, в настройках компонентов, или в вёрстке шаблона компонента, что сэкономит много времени