

Экзамен по созданию проектов на базе Bitrix Framework

## №2 «Основные инструменты кастомизации»

---

**Обновите документ, актуальная версия по ссылке: <https://academy.1c-bitrix.ru/~ex21desc>**

### Оглавление

Входные требования.....	2
Организационные правила .....	2
Условия для зачета решений .....	3
Предмет экзамена.....	4
Требования к решениям.....	6
Материалы для подготовки .....	9

## **Входные требования**

- Навыки реализации проектов в стеке PHP/MySQL/HTML/CSS/JS с использованием ООП.
- Сертификат по экзамену №1 «Интеграция дизайна и настройка публичной части» или «Интеграция дизайна и настройка платформы».

## **Организационные правила**

Для обеспечения объективности экзамена применяются определенные правила. Такой подход создает и защищает ценность полученного сертификата. **При нарушении правил экзамен засчитается как не пройденный, оплата не возвращается.**

### **Формат**

- Экзамен проходит удаленно, длительность — 4 часа.
- Потребуется отдельное помещение без присутствия кого-либо, кроме сдающего экзамен (дети допускаются). При необходимости допускается покидать помещение на время не более 10 минут за один раз.
- Необходим компьютер с подключением к сети Интернет, веб-камерой, операционной системой со всеми обновлениями: Windows 10, Windows 11 или MacOS последней версии.
- Нельзя использовать более одного монитора (отключить физически на время экзамена остальные). Если используется ноутбук с подключенным внешним монитором, то ноутбук должен быть закрыт.
- **Нельзя использовать наушники, общаться с кем-либо, производить запись экрана, делиться полученными заданиями.**
- Необходимо установить и настроить ПО для контроля процесса и подключения к серверу. **Система контроля обязательно должна работать в течение всего экзамена, обеспечить это - ответственность сдающего экзамен.** Подробная инструкция на сайте <https://academy.1c-bitrix.ru/certification/programs/>
- Результаты будут доступны в течение 10 рабочих дней на странице <http://academy.1c-bitrix.ru/personal/sertification/>, о чём придет уведомление на email. Сертификат скачивается в разделе «Мои сертификаты» <http://dev.1c-bitrix.ru/learning/mylearning.php#tab-certif-link>
- В случае несогласия с результатами необходимо в течение 5 рабочих дней создать обращение по ссылке <https://academy.1c-bitrix.ru/personal/support/>.

### **Справочные материалы**

Во время экзамена можно пользоваться только этими ресурсами:

- Справочником по PHP размещенным на рабочем столе экзаменационного сервера.

- Справочниками API на сайте: [https://dev.1c-bitrix.ru/api\\_d7/](https://dev.1c-bitrix.ru/api_d7/) и [https://dev.1c-bitrix.ru/api\\_help/](https://dev.1c-bitrix.ru/api_help/)
- Документацией по платформе: [https://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=43](https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43)

«Гуглить» и копировать готовые решения, использовать «свои заготовки», шаблоны кода в IDE, пользоваться помощью генеративных нейронных сетей – **запрещено**.

Разработчик, конечно, должен уметь находить готовые решения, а не изобретать велосипеды, использовать различные современные инструменты для быстрого решения задач, но это не применимо в рамках экзамена. Если можно будет «гуглить и копипастить ответы», то будет невозможно оценить личные знания разработчика.

Задания экзамена не подразумевают «головоломок», для успешного решения достаточно знаний в рамках предмета экзамена, опыта применения инструментов кастомизации.

## **Выделенный экзаменационный сервер**

Задания решаются на выделенном сервере с подключением к удаленному рабочему столу. Все действия, включая получение билета и работу с документацией на сайте dev.1c-bitrix.ru, происходят только на удалённом рабочем столе. Сервер обеспечивает равные условия для всех участников и контролируемый процесс экзамена.

Чтобы непривычная среда не отвлекала во время экзамена, ознакомьтесь заранее с настройками окружения <https://academy.1c-bitrix.ru/certification/programs/exam-server.php> и используйте образ демо-сервера <https://academy.1c-bitrix.ru/certification/programs/demo-server.php> во время подготовки.

Будьте готовы, что отзывчивость интерфейса удаленного рабочего стола уступает локальной. Это не медленная работа сервера, это задержка передачи изображения. Чем больше расстояние от вас до сервера, тем больше ping и более выражена задержка. Используйте рекомендации по оптимизации подключения <https://academy.1c-bitrix.ru/certification/programs/connection.php>.

## **Условия для зачета решений**

**Для зачета решений обязательно выполнение трех условий:**

### **1. Все задания из билета должны быть решены**

Объем заданий рассчитан на решения с учетом удаленного доступа, времени на самопроверку и запасом на возможные переподключения к серверу.

Методика проведения экзамена подразумевает, что сдающий знает необходимые технологии и имеет практику решения подобных задач. Разбираться во время экзамена как решается задача – не хватит времени.

Конкретные API не нужно помнить наизусть, во время экзамена доступна документация.

## **2. Соблюдены требования к решению**

Решить задания необходимо не только с точки зрения пользователя, но и соблюдая ряд технологических требований. Перечень требований указан в этом документе, им можно пользоваться во время экзамена.

В тексте заданий требования не дублируются, подразумевается, что сдающий познакомится с ними в процессе подготовки к экзамену и выучит их.

## **3. Фактическое решение на сервере работает верно**

Оценивается только фактический результат настроек и программирования, сохраненный на сервере после завершения экзамена.

Если что-то не работает из-за опечатки или при решении следующих заданий настройки или код предыдущих решений были перетерты – такое решение не будет зачтено, как и экзамен в целом.

У сдающего не будет возможности после экзамена объяснить, что он опечатался или он знал, но забыл во время экзамена, как именно решается задача или что-то случайно удалил.

Проверяйте итог перед завершением экзамена.

## **Предмет экзамена**

В рамках этого экзамена оцениваются знания, правильный выбор и навык применения основных инструментов кастомизации Bitrix Framework для решения задач, выходящих за рамки типового функционала системы.

### **1. Базовые элементы Bitrix Framework для решения задач кастомизации**

- Собственные модули для организации кода в рамках проекта.
- Страница и порядок её выполнения. Использование обработчиков событий из порядка выполнения страницы. Решение задач с учетом доступности объектов и буферизации контента на различных этапах выполнения страницы.
- Использование специальных констант и переменных.
- Настройка параметров ядра с помощью файла /bitrix/.settings.php
- Информационные блоки. Использование API для базового набора операций (CRUD).
- Пользователи и группы пользователей. Использование API для базового набора операций (CRUD).
- Использование в API значений типа список, дата/время, файл.
- Кастомизация стандартных языковых фраз в административном разделе.
- Административные инструменты:

- Создание записей в журнале событий с помощью API.

## **2. Расширение возможностей типовых компонентов**

- Использование parameters.php, result\_modifier.php, component\_epilog.php для расширения возможностей типовых компонентов.
- Дополнение данных для отображения в шаблоне компонента, кеширование дополнительных данных.
- Создание разного кеша компонента в зависимости от значений дополнительных данных.
- Вывод контента из компонента в любую точку страницы сайта.
- Задание значений метаданных страницы.

## **3. Обработчики событий**

- Регистрация обработчиков событий в модуле и на хитах.
- Использование обработчиков старого и нового ядра, addEventHandlerCompatible.
- Использование обработчиков событий главного модуля, информационных блоков, поиска, почтовой системы и других базовых модулей:
  - Изменить данные при создании или обновлении объекта, отменить действие.
  - Выполнить дополнительные действие в успешного завершения изменения или создания объекта.
  - Задать параметры для формирования контента страницы с помощью событий, вызываемых при исполнении страницы.
  - Повлиять на уже сгенерированный контент страницы и ее метаданные с помощью событий, вызываемых при исполнении страницы.
  - Кастомизировать административный раздел.
  - Повлиять на содержание и реквизиты письма при отправке почты.
  - Повлиять на индексацию данных для поиска.
  - Обработать сразу два события, и OnBefore и OnAfter, для выполнения дополнительный действий, только если данные успешно изменились необходимым образом и не было отмены обновления или создания объекта.

## **4. Кастомизация контента и метаданных страницы сайта**

- Указание значений свойств и метаданных страницы сайта с помощью отложенных функций.

- Использование пар функций Get\* и Set\* для изменения свойств и метаданных страницы.
- Использование AddViewContent/ShowViewContent для вывода контента в произвольной точке.
- Использование SetViewTarget/EndViewTarget/AddViewContent для вывода контента в произвольной точке.
- Создание собственных отложенных функций, AddBufferContent.

## 5. Дополнительные поля и свойства объектов

- Создание пользовательских полей (UF). Использование их в API (чтение, обновление, фильтрация по ним).
- Создание пользовательских свойств для элементов инфоблока. Использование их в API (чтение, обновление, фильтрация по ним).
- Особенности работы через API со свойствами типа список.

## 6. Почтовые уведомления

- Почтовые события, шаблоны, темы.
- Создание собственного почтового события и шаблона.
- Примерение Send и SendImmediate, таблицы b\_event.
- Обработка события отправки писем.

## 7. Агенты

- Параметры агента: функция, интервал и тип.
- Реализации функции агента.
- Ограничения в программном коде агентов.
- Выполнение агентов на хитах и на cron.

# Требования к решениям

## Общие:

- Ядро продукта не изменяется (стандартные компоненты, модули и т.д.).
- В рамках этого экзамена создание нового компонента или подмена стандартного через local будет неверным решением, компоненты должны использоваться типовые. Для решения

заданий необходимо использовать кастомизацию шаблонов стандартных компонентов и другие инструменты.

- Избегайте «бездумного копипаста». В стандартных шаблонах или примерах из документации могут примеры, которые помогут в решении заданий. Необходимо осмысленно применять готовый код, ваше решение должно соответствовать требованиям экзамена и условиям конкретной задачи. Если в решении присутствует лишний программный код или обработка данных, не требующихся для задачи, оно не будет засчитано.

### **Оформление кода:**

- Можно использовать API как «старого», так и «нового» ядра D7.
- Файлы с вашими объектами размещаются в папке /local/.
- Шаблон сайта и файл template.php компонента являются элементами «View» в схеме MVC, в них не размещается код, непосредственно реализующий бизнес-логику.
- ID информационных блоков, групп пользователей, свойств и т.д. не указываются в коде в явном виде, для них необходимо определять константы или переменные в одном месте для дальнейшего применения во всем проекте.
- Если необходимо передать данные между вызовами обработчиков событий или в целом в рамках хита – используйте, например, статическое свойство вашего класса или полноценный паттерн «Одиночка» (Singleton). Использование суперглобального массива \$GLOBALS устарело.

### **Текстовые фразы:**

- Текстовые фразы не пишутся явно вместе с PHP-кодом, не в HTML-шаблонах. Их необходимо выносить в языковые файлы, в том числе при использовании init.php и подключении в нём своих файлов/классов, в собственных модулях.

### **Базовая безопасность при выводе данных:**

- Вывод данных в шаблонах компонентов осуществляется из массива с ключами без «~». В ключах с «~» данные не приведены к безопасному виду, что, например, позволяет применить XSS.
- Используйте API, возвращающее данные, приведенные к безопасному виду для вывода в шаблонах.

### **Получение данных:**

- По умолчанию отбирайте только активные элементы инфоблоков, пользователей и других сущностей с типовым полем «Активность», если в задании явно не сказано другое.

- Прежде чем получать данные в API (GetList, GetByID и т.д.) с фильтром по какому-либо значению, необходимо убедиться, что оно не пустое, иначе можно получить ошибочную выборку данных.
- Получайте минимально необходимый набор данных для решения задачи, экономьте ресурсы сервера.
- Реализуйте алгоритмы, не приводящие к запросам к базе данных в цикле, если это возможно. Получение данных через API с выборкой по нескольким однотипным элементам необходимо реализовать через один вызов API с соответствующим фильтром и алгоритмом на уровне PHP, а не через вызов API в цикле на каждый элемент.
- Если объем получаемых данных можно ограничить на уровне выборки в вызове API, то следует так и сделать, а не выбирать данные с «запасом» и фильтровать затем в PHP.
- Не получайте одни и те же данные повторно через API, если при этом выполняется запрос в базу данных. Сохраните в PHP ранее полученные значения и используйте их для решения задачи.

#### **Обработчики событий:**

- Для передачи данных между вызовами обработчиков нельзя использовать массив arFields, передаваемый в обработчик по ссылке. На его содержимое может повлиять неожиданным для вас образом обработчики события, созданные не вами.
- Текстовые фразы выносите из кода обработчика события в языковые файлы.

#### **Кеширование в компонентах:**

- В настройках платформы и компонентах должно быть включено кеширование.
- При доработке компонентов получение дополнительной информации для вывода в шаблоне должно кешироваться.
- При кастомизации компонентов необходимо добавлять в кеш стандартного компонента через setResultCacheKeys значения только тех переменных, которые будут использоваться для решения задачи в некешируемой части компонента, чтобы не увеличивать размер кеша без необходимости.
- Применение отложенных функций при кастомизации компонентов должно учитывать работу механизма кеширования. Все доработки должны работать верно как при включенном, так и выключенном кешировании компонента.
- При решении задач по получению дополнительных данных в рамках шаблона типового компонента **не используйте** ключ **cache** метода **getList** ORM сущностей. Одновременная работа кеширования выборки в ORM и кеширования в компоненте приведет к конфликтам. Для выполнения задач этого экзамена достаточно воспользоваться стандартным механизмом кеширования компонента.

# **Материалы для подготовки**

## **Задания**

- Доступен пул заданий для подготовки. На экзамене в билете будет несколько заданий, аналогичных заданиям из пула.  
Скачать по ссылке: <https://academy.1c-bitrix.ru/~ex21train>
- Для оценки объема заданий во время экзамена доступен тестовый билет.  
Скачать по ссылке: <https://academy.1c-bitrix.ru/~ex21ticket>

## **Курсы**

- «Кастомизация публичной части»  
[https://academy.1c-bitrix.ru/courses/?COURSE\\_ID=304](https://academy.1c-bitrix.ru/courses/?COURSE_ID=304)
- «Основные технологии Bitrix Framework»  
[https://academy.1c-bitrix.ru/courses/?COURSE\\_ID=310](https://academy.1c-bitrix.ru/courses/?COURSE_ID=310)

## **Документация**

- «Разработчик Bitrix Framework» / «Middle, Средний уровень подготовки»  
[https://dev.1c-bitrix.ru/learning/course/index.php?COURSE\\_ID=43&CHAPTER\\_ID=03516&LESSON\\_PATH=3913.3516](https://dev.1c-bitrix.ru/learning/course/index.php?COURSE_ID=43&CHAPTER_ID=03516&LESSON_PATH=3913.3516)
- Порядок выполнения страницы  
[https://dev.1c-bitrix.ru/api\\_help/main/general/pageplan.php](https://dev.1c-bitrix.ru/api_help/main/general/pageplan.php)
- События главного модуля  
[https://dev.1c-bitrix.ru/api\\_help/main/events/index.php](https://dev.1c-bitrix.ru/api_help/main/events/index.php)
- API для работы с информационными блоками  
[https://dev.1c-bitrix.ru/api\\_help/iblock/index.php](https://dev.1c-bitrix.ru/api_help/iblock/index.php)
- API для работы с пользователями  
[https://dev.1c-bitrix.ru/api\\_help/main/reference/cuser/index.php](https://dev.1c-bitrix.ru/api_help/main/reference/cuser/index.php)