

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ПО ДИСЦИПЛИНЕ**  
Прикладное программирование на РНР

**Код модуля**  
1153122(2)

**Модуль**  
Прикладное программирование на РНР

**Екатеринбург**

Оценочные материалы составлены автором(ами):

<b>№ п/п</b>	<b>Фамилия, имя, отчество</b>	<b>Ученая степень, ученое звание</b>	<b>Должность</b>	<b>Подразделение</b>
1	Аксенов Константин Александрович	кандидат технических наук, доцент	Доцент	Департамент информационных технологий и автоматике
2	Спиричева Наталия Рахматулловна	без ученой степени, без ученого звания	Старший преподаватель	Департамент информационных технологий и автоматике

**Согласовано:**

Управление образовательных программ

Т.Г. Комарова

**Авторы:**

- Аксенов Константин Александрович, Доцент, Департамент информационных технологий и автоматике
- Спиричева Наталия Рахматулловна, Старший преподаватель, Департамент информационных технологий и автоматике

**1. СТРУКТУРА И ОБЪЕМ ДИСЦИПЛИНЫ Прикладное программирование на РНР**

1.	Объем дисциплины в зачетных единицах	3	
2.	Виды аудиторных занятий	Лекции Лабораторные занятия	
3.	Промежуточная аттестация	Зачет	
4.	Текущая аттестация	Домашняя работа	1

**2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ (ИНДИКАТОРЫ) ПО ДИСЦИПЛИНЕ МОДУЛЯ Прикладное программирование на РНР**

Индикатор – это признак / сигнал/ маркер, который показывает, на каком уровне обучающийся должен освоить результаты обучения и их предъявление должно подтвердить факт освоения предметного содержания данной дисциплины, указанного в табл. 1.3 РПМ-РПД.

Таблица 1

Код и наименование компетенции	Планируемые результаты обучения (индикаторы)	Контрольно-оценочные средства для оценивания достижения результата обучения по дисциплине
1	2	3
ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и	З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и	Домашняя работа Зачет Лабораторные занятия Лекции

<p>пользователям (Информатика и вычислительная техника)</p>	<p>комбинированной среды управления проектами Redmine и аналогов У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p>	
<p>ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и пользователям (Прикладная информатика)</p>	<p>З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и комбинированной среды управления проектами Redmine и аналогов У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p>	<p>Домашняя работа Зачет Лабораторные занятия Лекции</p>
<p>ПК-1 -Способен проектировать, разрабатывать, интегрировать, проверять на работоспособность программное обеспечение (модули, компоненты, продукты) и осуществлять разработку технических документов, адресованных специалисту по информационным технологиям и</p>	<p>З-2 - Различать синтаксис языков программирования, особенности программирования на этих языках, стандартные библиотеки языков программирования П-2 - Создавать и оптимизировать программный код на языках программирования высокого и низкого уровня с использованием специализированных программных средств П-3 - Иметь практический опыт использования инструмента контроля версий Git (GitHub, GitLab и др.) и</p>	<p>Домашняя работа Зачет Лабораторные занятия Лекции</p>

<p>пользователям (Программная инженерия)</p>	<p>комбинированной среды управления проектами Redmine и аналогов У-2 - Выбирать языки программирования для написания программного кода с учетом технического задания</p>	
<p>ПК-2 -Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования и анализировать результаты тестирования программного обеспечения (Информатика и вычислительная техника)</p>	<p>З-1 - Изложить теорию тестирования (модели тестирования, планирование тестирования, тест-дизайн, проектирование тестов и др.) З-2 - Перечислить основные техники тестирования, стандарты в области тестирования, методологии, применяемые к необходимым приложениям П-1 - Проводить необходимые виды тестирования в соответствии с планом тестирования П-2 - Выполнять анализ полученных результатов тестирования и оформлять в соответствии с требуемым форматом П-3 - Имеет практический опыт работы с тестовыми средами и системами управления тестированием (Test Link и аналоги) в своей профессиональной деятельности У-1 - Идентифицировать цели, объекты, входные данные и виды тестирования (приемочное, установочное, альфа- и бета-тестирование и др.) У-2 - Оценивать важность (приоритет выполнения) различных тестов (на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки)</p>	<p>Домашняя работа Зачет Лабораторные занятия Лекции</p>
<p>ПК-2 -Способен разрабатывать тесты, подготавливать тестовые данные,</p>	<p>З-1 - Изложить теорию тестирования (модели тестирования, планирование</p>	<p>Домашняя работа Зачет Лабораторные занятия Лекции</p>

<p>проводить тестирование, разрабатывать документы для тестирования и анализировать результаты тестирования программного обеспечения (Прикладная информатика)</p>	<p>тестирования, тест-дизайн, проектирование тестов и др.)  З-2 - Перечислить основные техники тестирования, стандарты в области тестирования, методологии, применяемые к необходимым приложениям  П-1 - Проводить необходимые виды тестирования в соответствии с планом тестирования  П-2 - Выполнять анализ полученных результатов тестирования и оформлять в соответствии с требуемым форматом  П-3 - Имеет практический опыт работы с тестовыми средами и системами управления тестированием (Test Link и аналоги) в своей профессиональной деятельности  У-1 - Идентифицировать цели, объекты, входные данные и виды тестирования (приемочное, установочное, альфа- и бета-тестирование и др.)  У-2 - Оценивать важность (приоритет выполнения) различных тестов (на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки)</p>	
<p>ПК-2 -Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования и анализировать результаты тестирования программного обеспечения</p>	<p>З-1 - Изложить теорию тестирования (модели тестирования, планирование тестирования, тест-дизайн, проектирование тестов и др.)  З-2 - Перечислить основные техники тестирования, стандарты в области тестирования, методологии, применяемые к необходимым приложениям  П-1 - Проводить необходимые виды тестирования в соответствии с планом тестирования</p>	<p>Зачет  Лабораторные занятия  Лекции</p>

(Программная инженерия)	<p>П-2 - Выполнять анализ полученных результатов тестирования и оформлять в соответствии с требуемым форматом</p> <p>П-3 - Имеет практический опыт работы с тестовыми средами и системами управления тестированием (Test Link и аналоги) в своей профессиональной деятельности</p> <p>У-1 - Идентифицировать цели, объекты, входные данные и виды тестирования (приемочное, установочное, альфа- и бета-тестирование и др.)</p> <p>У-2 - Оценивать важность (приоритет выполнения) различных тестов (на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки)</p>	
-------------------------	---	--

### 3. ПРОЦЕДУРЫ КОНТРОЛЯ И ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ В РАМКАХ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ МОДУЛЯ В БАЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ (ТЕХНОЛОГИЧЕСКАЯ КАРТА БРС)

#### 3.1. Процедуры текущей и промежуточной аттестации по дисциплине

<b>1. Лекции: коэффициент значимости совокупных результатов лекционных занятий – 0.50</b>		
Текущая аттестация на лекциях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах
<i>домашняя работа</i>	4,8	100
<b>Весовой коэффициент значимости результатов текущей аттестации по лекциям – 0.50</b>		
<b>Промежуточная аттестация по лекциям – зачет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по лекциям – 0.50</b>		
<b>2. Практические/семинарские занятия: коэффициент значимости совокупных результатов практических/семинарских занятий – не предусмотрено</b>		
Текущая аттестация на практических/семинарских занятиях	Сроки – семестр, учебная неделя	Максимальная оценка в баллах

<b>Весовой коэффициент значимости результатов текущей аттестации по практическим/семинарским занятиям – не предусмотрено</b>		
<b>Промежуточная аттестация по практическим/семинарским занятиям – нет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по практическим/семинарским занятиям – не предусмотрено</b>		
<b>3. Лабораторные занятия: коэффициент значимости совокупных результатов лабораторных занятий – 0.50</b>		
<b>Текущая аттестация на лабораторных занятиях</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<i>выполнение и защита лабораторных работ</i>	4,16	100
<b>Весовой коэффициент значимости результатов текущей аттестации по лабораторным занятиям – 1.00</b>		
<b>Промежуточная аттестация по лабораторным занятиям – нет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по лабораторным занятиям – 0.00</b>		
<b>4. Онлайн-занятия: коэффициент значимости совокупных результатов онлайн-занятий – не предусмотрено</b>		
<b>Текущая аттестация на онлайн-занятиях</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<b>Весовой коэффициент значимости результатов текущей аттестации по онлайн-занятиям – не предусмотрено</b>		
<b>Промежуточная аттестация по онлайн-занятиям – нет</b>		
<b>Весовой коэффициент значимости результатов промежуточной аттестации по онлайн-занятиям – не предусмотрено</b>		

### 3.2. Процедуры текущей и промежуточной аттестации курсовой работы/проекта

<b>Текущая аттестация выполнения курсовой работы/проекта</b>	<b>Сроки – семестр, учебная неделя</b>	<b>Максимальная оценка в баллах</b>
<b>Весовой коэффициент текущей аттестации выполнения курсовой работы/проекта – не предусмотрено</b>		
<b>Весовой коэффициент промежуточной аттестации выполнения курсовой работы/проекта – защиты – не предусмотрено</b>		

## 4. КРИТЕРИИ И УРОВНИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ МОДУЛЯ

4.1. В рамках БРС применяются утвержденные на кафедре/институте критерии (признаки) оценивания достижений студентов по дисциплине модуля (табл. 4) в рамках контрольно-оценочных мероприятий на соответствие указанным в табл.1 результатам обучения (индикаторам).

Таблица 4

### Критерии оценивания учебных достижений обучающихся

<b>Результаты обучения</b>	<b>Критерии оценивания учебных достижений, обучающихся на соответствие результатам обучения/индикаторам</b>
----------------------------	---



Знания	Студент демонстрирует знания и понимание в области изучения на уровне указанных индикаторов и необходимые для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Умения	Студент может применять свои знания и понимание в контекстах, представленных в оценочных заданиях, демонстрирует освоение умений на уровне указанных индикаторов и необходимых для продолжения обучения и/или выполнения трудовых функций и действий, связанных с профессиональной деятельностью.
Опыт /владение	Студент демонстрирует опыт в области изучения на уровне указанных индикаторов.
Другие результаты	Студент демонстрирует ответственность в освоении результатов обучения на уровне запланированных индикаторов. Студент способен выносить суждения, делать оценки и формулировать выводы в области изучения. Студент может сообщать преподавателю и коллегам своего уровня собственное понимание и умения в области изучения.

4.2 Для оценивания уровня выполнения критериев (уровня достижений обучающихся при проведении контрольно-оценочных мероприятий по дисциплине модуля) используется универсальная шкала (табл. 5).

Таблица 5

#### Шкала оценивания достижения результатов обучения (индикаторов) по уровням

Характеристика уровней достижения результатов обучения (индикаторов)				
№ п/п	Содержание уровня выполнения критерия оценивания результатов обучения (выполненное оценочное задание)	Шкала оценивания		
		Традиционная характеристика уровня		Качественная характеристика уровня
1.	Результаты обучения (индикаторы) достигнуты в полном объеме, замечаний нет	Отлично (80-100 баллов)	Зачтено	Высокий (В)
2.	Результаты обучения (индикаторы) в целом достигнуты, имеются замечания, которые не требуют обязательного устранения	Хорошо (60-79 баллов)		Средний (С)
3.	Результаты обучения (индикаторы) достигнуты не в полной мере, есть замечания	Удовлетворительно (40-59 баллов)		Пороговый (П)
4.	Освоение результатов обучения не соответствует индикаторам, имеются существенные ошибки и замечания, требуется доработка	Неудовлетворительно (менее 40 баллов)	Не зачтено	Недостаточный (Н)
5.	Результат обучения не достигнут, задание не выполнено	Недостаточно свидетельств для оценивания		Нет результата

## 5. СОДЕРЖАНИЕ КОНТРОЛЬНО-ОЦЕНОЧНЫХ МЕРОПРИЯТИЙ ПО ДИСЦИПЛИНЕ МОДУЛЯ

### 5.1. Описание аудиторных контрольно-оценочных мероприятий по дисциплине модуля

#### 5.1.1. Лекции

Самостоятельное изучение теоретического материала по темам/разделам лекций в соответствии с содержанием дисциплины (п. 1.2. РПД)

#### 5.1.2. Лабораторные занятия

Примерный перечень тем

1. Создание базового проекта с помощью фреймворка Laravel  
2. Реализация концепции MVC. Настройка моделей, роутов и контроллеров с помощью фреймворка Laravel

3. Реализация концепции MVC. Настройка представлений с помощью фреймворка Laravel

LMS-платформа – не предусмотрена

### 5.2. Описание внеаудиторных контрольно-оценочных мероприятий и средств текущего контроля по дисциплине модуля

Разноуровневое (дифференцированное) обучение.

#### Базовый

#### 5.2.1. Домашняя работа

Примерный перечень тем

1. Реализация backend-приложения с использованием современных фреймворков на языке php

Примерные задания

Подготовка

1. Убедитесь, что у вас есть аккаунт в какой-либо системе удаленных репозиторий, например, в GitHub (<https://github.com/>) или GitLab (<https://gitlab.com/>). Зарегистрируйтесь при необходимости.

2. Создайте новый репозиторий в своем пространстве.

a. Для github <https://github.com/new>

b. Для gitlab <https://gitlab.com/projects/new>

3. Настройте клиент Git для работы. При необходимости обращайтесь к заданиям лабораторной работы №3 предыдущего курса.

4. Настройте окружение для работы – вам нужен работающая СУБД, web-сервер и интерпретатор PHP. Пользователям windows для удобства рекомендуется пользоваться пакетом OpenServerPanel.

Предметная область

Тематику и предметную область предлагается выбрать самостоятельно.

1. Распишите небольшое описание будущей системы или ТЗ на разработку. Согласуйте с преподавателями по бригадам.

2. Продумайте структуру базы данных. Опишите в удобной нотации.

Создание проекта

1. Выберите фреймворк для удобной работы. Предлагается использовать Laravel (<https://laravel.com/>). Дальнейшие объяснения заложены под него. Если в используемом вами инструменте требуемый функционал реализован иначе или отсутствует – реализуйте цели другими доступными способами.

2. Создайте пустой проект Laravel. Следуйте указаниям в документации (<https://laravel.com/docs/8.x/installation>).

Рекомендуется воспользоваться установкой с помощью менеджера зависимостей Composer (убедитесь, что он у вас есть).

3. Создайте на основе файла `.env.example` файл `.env`

Это файл настроек вашего окружения и конфигурационных переменных, для хранения доступов, паролей и многого другого. Работая дальше помните, что этот файл не должен попасть в ваш удаленный репозиторий НИ ПРИ КАКИХ УСЛОВИЯХ.

Создание локального git репозитория

1. Проинициализируйте локальный репозиторий.

2. Создайте и наполните файлы `gitignore` (вот как раз например `.env`).

3. Добавьте файлы в репозиторий и зафиксируйте изменения.

4. Подключите удаленный git репозиторий.

5. Отправьте свои первые изменения в мастер ветку.

6. Создайте новую ветку. Можете приступать к работе.

Настройка проекта для локальной работы

1. Сгенерируйте ключ приложения.

2. Пропишите в `.env` данные для подключения к базе данных.

3. Ознакомьтесь с другими базовыми настройками, вынесенными разработчиками в `.env.example`. Что они означают? При необходимости настройте отдельные параметры.

Реализуем MVC

Модели и миграции

1. Начнём с моделей. Создайте модели и миграции для ваших сущностей базы данных, которые вы продумывали ранее.

2. В миграциях пропишите названия таблиц, поля, типы данных, значения по умолчанию и т.п.

Помните, что миграции НЕЛЬЗЯ редактировать, если вы уже отправили свои изменения в удаленный репозиторий. Для внесения изменений используются новые миграции.

3. Давайте продемонстрируем это. Упустите намеренно описание какого-либо поля в какой-либо таблице. Создайте миграцию, которая добавит это поле.

4. Опишите связи в моделях (`belongsTo`, `hasMany`, `belongsToMany`, etc).

5. Опишите уровень защиты полей моделей (`fillable`, `guarded`).

6. Опишите преобразование типов для работы с БД в моделях при необходимости (`casts`).

Контроллеры

1. Создайте контроллеры для реализации CRUD для ваших моделей – это нам пригодится для роли администратора.

2. Реализуйте методы create, update, index и delete. Вы реализуете api, так что позаботьтесь об эффективной сериализации ответа в json.
  3. Дополнительно. Реализуйте для метода index постраничную пагинацию данных. Можете использовать сторонние библиотеки, установленные через composer. Сейчас это делать необязательно, но потом для проекта это всё равно будет нужно.
  4. Дополнительно. Реализуйте для метода index фильтрацию и сортировку по различным полям. Можете использовать сторонние библиотеки, установленные через composer. Сейчас это делать необязательно, но потом для проекта это всё равно будет нужно.
  5. Дополнительно. Реализуйте концепцию soft deletes, чтобы при удалении ваши данные фактически никогда не исчезали из самой базы данных, но api не возвращало их в своих ответах.
  6. Дополнительно. Реализовывать систему ролей и систему защиты доступа – продемонстрируете это при реализации проекта. Рекомендуется использовать для этого посредников и стражей (middlewares и guards), или готовые решения, которые вы установите с помощью composer.
  7. Настройте роуты для вызова методов вашего контроллера.
- Заканчиваем
- Запустите изменения и выполните слияние ветки через Pull Request / Merge Request. Можете продолжать развивать проект самостоятельно.
- Реализуем проект
- Мы попробовали весь базисный функционал. Все необходимые модели, контроллеры, представления или сериализацию в json для API (рекомендуется использовать Recourses), посредников, роуты и т.п.
- Тестирование
- Покройте реализованный функционал тестами

LMS-платформа – не предусмотрена

### **5.3. Описание контрольно-оценочных мероприятий промежуточного контроля по дисциплине модуля**

#### **5.3.1. Зачет**

Список примерных вопросов

1. Енамы. Назначение и способы реализации
2. Трейты. Назначение и способы реализации
3. Назначение CMS. Особенности. Отличия от полноценной разработки. Принципы No Code
4. MVC. Принципы. Особенности архитектуры
5. MVC. Уровень ответственности контроллеров
6. MVC. Отличия моделей и ORM
7. SOLID. Принципы
8. Виды тестирования. Юнит тесты
9. Виды тестирования. Функциональные тесты
10. Автоматизация тестирования. Способы. Реализация для PHP

11. Фреймворки. Назначение. Особенности. Примеры
  12. Проблемно-ориентированное проектирование. Основные принципы
  13. Проблемно-ориентированное проектирование. Уровни абстракции
  14. Методологии работы над проектом. Scrum. Принципы. Особенности
  15. Методологии работы над проектом. Waterfall. Принципы. Особенности
  16. Методологии работы над проектом. Kanban. Принципы. Особенности
  17. Принципы командной работы над проектом. Системы контроля версий. Диаграммы Ганта
- LMS-платформа – не предусмотрена

#### 5.4 Содержание контрольно-оценочных мероприятий по направлениям воспитательной деятельности

Направление воспитательной деятельности	Вид воспитательной деятельности	Технология воспитательной деятельности	Компетенция	Результаты обучения	Контрольно-оценочные мероприятия
Профессиональное воспитание	целенаправленная работа с информацией для использования в практических целях	Технология формирования уверенности и готовности к самостоятельной успешной профессиональной деятельности Технология самостоятельной работы	ПК-1	П-3	Зачет Лабораторные занятия Лекции
			ПК-1	П-3	
			ПК-1	П-3	
			ПК-1	П-3	
			ПК-1	П-3	
			ПК-1	П-3	
			ПК-1	П-3	