

Документ подписан простой электронной подписью

Информационные материалы для промежуточной аттестации по дисциплине

ФИО: Косенок Сергей Михайлович

Должность: ректор

Дата подписания: 24.06.2026 06:57:39

Уникальный программный ключ:

e3a68f3eaa1e62674b54f4998099d3d6bdfcf836

Разработка WEB-приложений

Код, направление подготовки	09.03.02 Информационные системы и технологии
Направленность (профиль)	Безопасность информационных систем и технологий
Форма обучения	Очная
Кафедра-разработчик	Информатики и вычислительной техники
Выпускающая кафедра	Информатики и вычислительной техники

Типовые задания для контрольной работы:

Задание 1

1. Реализуйте класс Eat, который имеет свойства type и name и позволяет получать их с помощью функций getName, getType, но изменять эти свойства позволяют лишь методы setName(\$name), setType(\$type).
2. Реализуйте связанные наследованием классы Man и Student, так, чтобы дочерний класс наследовал от родительского свойство \$age, и при этом оно было доступно через объекты дочернего класса, но недоступно через объекты базового класса. 2. Реализуйте метод getAge() в классе Man, так чтобы этот метод возвращал значение свойства \$age.

```
<?php
class Man{
    protected $age;
    public function getAge()
    {
        return $this->age;
    }
}
class Student extends Man{
    public $age;
}
$man1 = new Man();
$stud1 = new Student();
```

Задание 2

1. Для следующего примера кода сделайте так, чтобы классы Project и Task гарантировано имели метод execute(\$executor), а метод setWork класса employee принимал любые объекты классов, которые имеют метод execute(\$executor)

```
<?php
class Project{
    private $id;
    private $title;
    private $description;
}
class Task{
    private $id;
    private $title;
```

```

    private $description;
}
class Employee{
    private $work;
    public function setWork($executable){
        $this->work = $executable;
        $this->work->execute($this);
    }
}
$employee = new Employee();
$employee->setWork(new Task());
$employee->setWork(new Project());

```

Задание 3

1. Определите, каким образом будет преобразован URL <http://сайт/admin> при помощи следующего .htaccess:

```

<IfModule mod_rewrite.c>
RewriteEngine on
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule (.*) /index.php?$1 [QSA,L]
</IfModule>

```

Задание 4

1. При помощи механизма PHP-сессий реализовать на сайте защищенный просмотр страниц и авторизацию через http-форму.
2. В меню категорий предусмотреть ссылку «Вход», при нажатии на которую пользователю предлагается ввести логин и пароль в стандартном окне браузера.
3. После прохождения аутентификации в меню категорий постоянно должна отображаться информация о том, что пользователь аутентифицирован и должен выводиться его логин.
4. При попытке доступа к страницам с информацией об объектах без прохождения аутентификации в окне объектов должно выводиться сообщение с предложением пройти аутентификацию и ссылка «Вход».

Типовые темы курсового проекта

1. Система управления задачами: Создание веб-приложения для управления проектами и задачами с возможностью назначения исполнителей и отслеживания статуса выполнения.
2. Онлайн-каталог товаров: Разработка интернет-магазина с фильтрацией и поиском по категориям товаров, возможностью добавления в корзину и оформления заказа.
3. Платформа для блогов: Создание платформы, на которой пользователи могут публиковать статьи, оставлять комментарии и подписываться на обновления.
4. Система опросов и анкетирования: Веб-приложение для создания и проведения опросов, анализа результатов и генерации отчетов.
5. Приложение для учета расходов: Разработка системы личных финансов с возможностью ввода доходов и расходов, составления отчетов и анализа финансовых привычек.
6. Образовательный портал: Веб-приложение для размещения курсов, материалов, тестов и взаимодействия между преподавателями и студентами.
7. Система управления событиями: Создание платформы для организации мероприятий, регистрации участников и отслеживания их посещаемости.

8. Форум для обсуждений: Разработка веб-форума с темами и категориями, где пользователи могут создавать новые темы и обсуждать их.
9. Веб-сервис для обмена фотографиями: Создание платформы для загрузки, хранения и обмена фотографиями с возможностью комментирования.
10. Система рекомендаций фильмов/книг: Веб-приложение, которое предлагает пользователям фильмы или книги на основе их предпочтений.
11. Чат-приложение в реальном времени: Разработка веб-приложения для общения между пользователями в режиме реального времени с использованием WebSocket.
12. Система управления запасами: Создание веб-приложения для управления инвентарем, учета товаров и составления отчетов.
13. Проект для планирования питания: Веб-приложение, которое предлагает рецепты и планирует питание на неделю с учетом предпочтений пользователя.
14. Система для управления резюме: Создание платформы для поиска работы, публикации резюме и взаимодействия со специалистами по рекрутингу.
15. Платформа для совместного изучения: Веб-приложение, где пользователи могут организовывать группы для изучения и совместной работы над проектами.
16. Система для мониторинга здоровья: Разработка веб-приложения для отслеживания состояния здоровья и ведения дневника активности.
17. Интерактивная карта с точками интереса: Создание приложения, которое отображает интересные места на карте с возможностью добавления пользовательских маркеров.
18. Система обратной связи: Веб-приложение для сбора мнений и отзывов от пользователей, с возможностью анализа и представления результатов.
19. Агрегатор новостей: Создание веб-приложения, которое собирает новости из различных источников и предоставляет пользователям возможность фильтровать по интересам.
20. Система закладок и аннотаций: Веб-приложение для сохранения и организации ссылок с возможностью добавления заметок и тегов.

Типовые вопросы к экзамену:

1. Дайте определение терминам а) инкапсуляция б) наследование
2. Дайте определение терминам а) полиморфизм б) абстрактный класс в) интерфейс
3. Дайте определение терминам а) статический класс б) статический метод в) статическое свойство.
4. Для каких целей используется паттерн Singleton?
5. Перечислите основные шаги реализации паттерна Singleton на языке PHP.
6. Дайте определение термину «магический метод».
7. Перечислите основные магические методы в языке PHP.
8. Как можно указать файл, открываемый web-сервером по умолчанию, если URL запроса содержит неверный путь к файлу?
9. Опишите принцип, каким образом backend-приложение использует информацию, содержащуюся в URL.
10. Опишите принципы организации контента с использованием сетки Bootstrap.
11. Опишите принципы применения CSS-свойства display: flex.
12. В чем заключается принцип адаптивной верстки страниц?
13. Опишите назначение и работу механизма HTTP cookies.
14. Опишите назначение и работу механизма PHP-сессий.
15. Приведите пример разграничения доступа пользователя к php-программе на сервере.
16. Опишите назначение и работу технологии AJAX.
17. В чем заключается разница между синхронным и асинхронным AJAX-запросом.

