

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ А.Г.Баханович

Регистрационный № _____

ВЫСОКОУРОВНЕВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

**Примерная учебная программа по учебной дисциплине
для специальности**

6-05-0611-01 Информационные системы и технологии

СОГЛАСОВАНО

Председатель Учебно-методического
объединения по образованию в
области информатики и
радиоэлектроники

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.Н. Пищов

СОГЛАСОВАНО

Проректор по научно-методической
работе Государственного учреждения
образования «Республиканский
институт высшей школы»

_____ И.В. Титович

Эксперт-нормоконтролер

СОСТАВИТЕЛИ:

Д.А.Сторожев, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», магистр экономических наук, исследователь технических наук;

Н.А.Атрощенко, старший преподаватель кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»;

В.Н.Комличенко, доцент кафедры экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

РЕЦЕНЗЕНТЫ:

Кафедра информационных технологий в экологии и медицине учреждения образования «Международный государственный экологический институт имени А.Д.Сахарова» Белорусского государственного университета (протокол № 10а от 16.05.2025);

Н.В.Шалькевич, директор общества с ограниченной ответственностью «Трисофт»

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой экономической информатики учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 18 от 30.06.2025);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол №__ от _____);

Научно-методическим советом по прикладным информационным системам и технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 1 от 02.09.2025)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Примерная учебная программа по учебной дисциплине «Высокоуровневые языки программирования» разработана для студентов учреждений высшего образования, обучающихся по специальности 6-05-0611-01 «Информационные системы и технологии» в соответствии с требованиями образовательного стандарта общего высшего образования и примерного учебного плана вышеуказанной специальности.

Актуальность изучения учебной дисциплины «Высокоуровневые языки программирования» обоснована необходимостью подготовки специалистов, владеющих фундаментальными знаниями и практическими навыками в области проектирования и создания программных продуктов с использованием современных инструментов и парадигм программирования. Освоение таких языков позволяет эффективно реализовывать алгоритмы, разрабатывать масштабируемые и надёжные программные решения, а также адаптироваться к постоянно меняющимся требованиям информационных технологий. Кроме того, знание высокоуровневых языков программирования является основой для успешной деятельности в области информационных систем, включая веб-разработку, мобильные приложения, встраиваемые системы и корпоративное программное обеспечение.

Воспитательное значение учебной дисциплины «Высокоуровневые языки программирования» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

ЦЕЛЬ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: приобретение теоретических знаний и практических навыков, необходимых для эффективной разработки программного обеспечения с использованием современных языков программирования высокого уровня.

Задачи учебной дисциплины:

изучение основных понятий, структур и средств высокоуровневого программирования;

освоение структурного, модульного и объектно-ориентированного подходов;

приобретение навыков разработки, тестирования и отладки программ;

подготовка к использованию современных языков (например, JavaScript, PHP, Python, Java, C#) в проектной деятельности;

приобретение навыков работы в интегрированной среде разработки при создании веб-приложений.

Базовыми учебными дисциплинами по курсу «Высокоуровневые языки программирования» являются «Основы алгоритмизации и программирования», «Объектно-ориентированное проектирование и программирование», «Программирование сетевых приложений». В свою очередь, учебная дисциплина «Высокоуровневые языки программирования» является базой для такой учебной дисциплины компонента учреждения образования, как «Распределенные информационные системы».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Высокоуровневые языки программирования» формируется следующая базовая профессиональная компетенция: применять средства и инструменты высокоуровневых языков программирования для реализации проектных решений в информационных системах.

В результате изучения учебной дисциплины студент должен:

знать:

основные принципы высокоуровневого программирования, его особенности и отличия от низкоуровневых языков;

синтаксис и семантику выбранных языков программирования (например, PHP, JavaScript, Java, C#, Python);

структуры данных и способы их применения в программировании;

парадигмы программирования: процедурное, объектно-ориентированное, функциональное;

методы обработки ошибок и исключительных ситуаций;

принципы модульности, повторного использования кода и работы с библиотеками;

основы работы со средами разработки и отладки программ;

этапы жизненного цикла программного обеспечения и роль программирования в нем;

уметь:

применять синтаксис и основные конструкции высокоуровневых языков программирования для решения типовых задач;

разрабатывать программные модули с использованием процедурного и объектно-ориентированного подходов;

использовать встроенные и подключаемые библиотеки и компоненты;

осуществлять тестирование, отладку и модификацию программных решений;

документировать исходный код в соответствии с современными стандартами;

разрабатывать небольшие приложения или фрагменты информационных систем, используя выбранные средства разработки;

применять среду разработки (IDE) для реализации и отладки программ;

эффективно использовать средства ввода-вывода и средства работы с данными (в т.ч. файлами, коллекциями, базами данных);

иметь навык:

разработки программ с использованием высокоуровневых языков программирования, включая объектно-ориентированный и структурный подход;

эффективного построения алгоритмов и структур данных для решения практических задач;

отладки и тестирования программного обеспечения с применением стандартных IDE и утилит;

документирования и сопровождения исходного кода в соответствии с современными стандартами;

интеграции программных модулей и использования сторонних библиотек и фреймворков;

взаимодействия с внешними данными, включая работу с файлами, базами данных и API;

организовать командную разработку программного продукта, используя системы контроля версий (например, Git).

Примерная учебная программа рассчитана на 108 учебных часов, из них – 52 аудиторных часа. Примерное распределение аудиторных часов по видам занятий: лекции – 24 часа, лабораторные занятия – 28 часов.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Введение в высокоуровневые языки программирования. Обзор парадигм	4	4	–
Тема 1. Понятие и классификация языков программирования. Обзор парадигм программирования. Особенности высокоуровневых языков	2	2	–
Тема 2. Современные тренды и области применения. Обзор сред разработки и выполнения программ	2	2	–
Раздел 2. Введение в web-разработку и основы синтаксиса высокоуровневого языка программирования	34	14	20
Тема 3. Введение в веб-разработку. Веб-архитектура и клиент-серверное взаимодействие. Введение в мультипарадигменный высокоуровневый язык программирования	2	2	–
Тема 4. Основные конструкции языка. Массивы. Управляющие структуры и организация циклов. Функции. Обработка событий	6	2	4
Тема 5. Особенности функционального программирования и работы с функциями. Исключения и обработка исключительных ситуаций	6	2	4
Тема 6. Особенности организации сложных структур данных. Работа с коллекциями. Встроенные структуры данных и подходы к их обработке	6	2	4
Тема 7. Объектно-ориентированное программирование. Понятие объекта и класса в языке программирования. Стандартные встроенные объекты и их назначение. Организация работы с объектами	6	2	4
Тема 8. Классы и наследование. Инкапсуляция, наследование, интерфейсы	6	2	4
Тема 9. Повышение эффективности работы веб-приложения за счёт хранения данных на клиентской стороне. Сессии, куки и хранение данных	2	2	–
Раздел 3. Приёмы, средства и инструменты для разработки клиент-серверных веб-приложений с источником данных	14	6	8
Тема 10. Организация модульного подхода при разработке проектов. Основы программирования на стороне сервера	2	2	–

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Тема 11. Организация работы с файлами и файловой системой. Работа с XML/JSON и обмен данными. Программная реализация сценариев на основе технологии Ajax	6	2	4
Тема 12. Работа с базами данных. Безопасность веб-приложений	6	2	4
Итого:	52	24	28

* Для разработки учебных программ учреждения образования по данной учебной дисциплине может использоваться любой высокоуровневый язык программирования.

Перечень популярных высокоуровневых языков: Python, Java, JavaScript, PHP, C#, Ruby, Swift, Kotlin, Go (Golang), Rust, TypeScript.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ВВЕДЕНИЕ В ВЫСОКОУРОВНЕВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ОБЗОР ПАРАДИГМ

Тема 1. ПОНЯТИЕ И КЛАССИФИКАЦИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ. ОБЗОР ПАРАДИГМ ПРОГРАММИРОВАНИЯ. ОСОБЕННОСТИ ВЫСОКОУРОВНЕВЫХ ЯЗЫКОВ

Предмет курса и содержание учебной дисциплины, ее связь с другими учебными дисциплинами. Классификация языков программирования (по уровню абстракции, по способу реализации, по назначению, по парадигмам). Обзор парадигм программирования: процедурная (C, Pascal), объектно-ориентированная (Java, C++, PHP, Python), функциональная (Haskell, Lisp, Elm), логическая (Prolog), событийно-ориентированная (JavaScript, VBScript), декларативная (SQL, HTML как частный случай). Особенности высокоуровневых языков. Сравнение языков разных парадигм. Примеры реализации одной задачи в разных стилях. Преимущества и ограничения подходов.

Тема 2. СОВРЕМЕННЫЕ ТРЕНДЫ И ОБЛАСТИ ПРИМЕНЕНИЯ. ОБЗОР СРЕД РАЗРАБОТКИ И ВЫПОЛНЕНИЯ ПРОГРАММ

Современные тренды в разработке приложений на высокоуровневых языках программирования. Области применения. Обзор сред разработки (IDE и редакторов). Среды выполнения программ. Инструменты и экосистемы. Современные подходы к разработке web-приложений. Развитие облачных технологий и DevOps. Безопасность и защита данных. Тестирование и отладка. Производительность и оптимизация. Тренды в UI/UX и фронтенде. Интеграция с внешними сервисами (REST и GraphQL API, Webhooks и события, интеграция с платёжными системами, картами, чатами и др.)

Раздел 2. ВВЕДЕНИЕ В WEB-РАЗРАБОТКУ И ОСНОВЫ СИНТАКСИСА ВЫСОКОУРОВНЕВОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

Тема 3. ВВЕДЕНИЕ В WEB-РАЗРАБОТКУ. WEB-АРХИТЕКТУРА И КЛИЕНТ-СЕРВЕРНОЕ ВЗАИМОДЕЙСТВИЕ. ВВЕДЕНИЕ В МУЛЬТИПАРАДИГМЕННЫЙ ВЫСОКОУРОВНЕВЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ

Понятие веб-разработки: цели, задачи, направления. Разделение на frontend и backend. Обзор жизненного цикла веб-приложения. Понятие клиент-серверной модели. HTTP-протокол: запросы и ответы, методы (GET, POST и др.). Роль браузера, сервера и базы данных. REST и архитектура API. Примеры взаимодействия: форма → сервер → ответ. Классификация, место, назначение, возможности и история развития высокоуровневого языка программирования. Версии, спецификации, редакторы языка. Справочники и документация. Синтаксис языка. Лексемы. Идентификаторы. Ключевые и зарезервированные

слова. Типы данных. Особенности типизации. Приведение типов данных. Константы. Переменные. Выражения. Комментарии. Команды ввода и вывода. Этапы разработки сценария. Структура кода. Скрипт и его синтаксис.

Тема 4. ОСНОВНЫЕ КОНСТРУКЦИИ ЯЗЫКА. МАССИВЫ. УПРАВЛЯЮЩИЕ СТРУКТУРЫ И ОРГАНИЗАЦИЯ ЦИКЛОВ. ФУНКЦИИ. ОБРАБОТКА СОБЫТИЙ

Одномерные и многомерные массивы. Ассоциативные массивы и объекты как словари. Методы и функции работы с массивами. Управляющие структуры: условные операторы, тернарный оператор, switch-конструкция и её особенности. Организация циклов: циклы for, while, do...while, прерывание и продолжение: break, continue; вложенные циклы и их применение, итерация по коллекциям и объектам. Объявление и вызов функций. Аргументы и значения по умолчанию. Возврат значений (return). Анонимные функции и замыкания (closures). Стрелочные функции. Рекурсия и её применение. Создание сценариев с использованием функций. Локальные и глобальные переменные. Области видимости. Виды событий. Обработчики событий.

Тема 5. ОСОБЕННОСТИ ФУНКЦИОНАЛЬНОГО ПРОГРАММИРОВАНИЯ И РАБОТЫ С ФУНКЦИЯМИ. ИСКЛЮЧЕНИЯ И ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ

Принципы функционального программирования: чистые функции (pure functions), неизменяемость данных (immutability), функции высшего порядка (higher-order functions), рекурсия вместо циклов, ссылочная прозрачность (referential transparency), отсутствие побочных эффектов. Каррирование и композиция функций. Понятие исключительной ситуации. Синхронные и асинхронные исключения. Механизмы обработки: try...catch, finally. Генерация исключений: throw, Exception, Error. Пользовательские исключения. Практика обработки ошибок. Обработка ошибок при работе с файлами, сетью, вводом/выводом. Логирование и отладка ошибок. Принципы fail-safe и graceful degradation. Связь между функциональным программированием и обработкой исключений: как чистые функции упрощают отладку, использование Either, Maybe, Result-подобных структур (в функциональных языках). Примеры безопасного кода без try...catch.

Тема 6. ОСОБЕННОСТИ ОРГАНИЗАЦИИ СЛОЖНЫХ СТРУКТУР ДАННЫХ. РАБОТА С КОЛЛЕКЦИЯМИ. ВСТРОЕННЫЕ СТРУКТУРЫ ДАННЫХ И ПОДХОДЫ К ИХ ОБРАБОТКЕ

Понятие и классификация структур данных. Императивный и декларативный подход к обработке массивов данных. Встроенные методы для обработки элементов массива на декларативной основе. Коллекции в высокоуровневых языках программирования: Array, Set, Map, WeakMap, WeakSet, ArrayObject, SplFixedArray, SplStack, SplQueue. Методы обработки: map, filter, reduce, forEach, array_map, array_filter. Подходы к обработке коллекций: императивный и декларативный стиль, использование итераторов и

генераторов, ленивые вычисления и потоки данных. Встроенные структуры данных и их особенности: стандартные библиотеки, преимущества использования встроенных структур, ограничения и расширения через пользовательские классы. Организация и хранение сложных структур: вложенные коллекции (массивы массивов, словари списков и т.д.), сериализация и десериализация (JSON, XML).

Тема 7. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ. ПОНЯТИЕ ОБЪЕКТА И КЛАССА В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ. СТАНДАРТНЫЕ ВСТРОЕННЫЕ ОБЪЕКТЫ И ИХ НАЗНАЧЕНИЕ. ОРГАНИЗАЦИЯ РАБОТЫ С ОБЪЕКТАМИ

Основы объектно-ориентированного программирования (ООП): понятие парадигмы ООП; преимущества ООП: модульность, повторное использование, масштабируемость; сравнение с процедурным и функциональным подходами. Понятие объекта в языке программирования. Объект как сущность с состоянием и поведением. Свойства (поля) и методы объекта. Экземпляр класса и его жизненный цикл. Стандартные встроенные объекты и их назначение. Создание, изменение и удаление объектов. Доступ к свойствам и методам. Работа с коллекциями объектов. Использование итераторов и генераторов. Принципы слабой связанности и высокой связности. Объектная модель в веб-разработке: объекты DOM, модели MVC и MVVM, использование ООП в фреймворках.

Тема 8. КЛАССЫ И НАСЛЕДОВАНИЕ. ИНКАПСУЛЯЦИЯ, НАСЛЕДОВАНИЕ, ИНТЕРФЕЙСЫ

Понятие класса как шаблона объекта. Создание экземпляров (объектов). Свойства (поля) и методы. Конструкторы и деструкторы. Инкапсуляция. Скрытие данных и уровни доступа (public, protected, private). Геттеры и сеттеры. Примеры инкапсуляции. Наследование. Базовый и производный классы. Переопределение методов (overriding). Использование parent::, super. Множественное наследование (возможности и ограничения). Примеры и иерархии классов. Преимущества: безопасность, контроль доступа, модульность. Интерфейсы и абстракция. Интерфейсы как контракт поведения. Абстрактные классы и методы. Различия между интерфейсами и абстрактными классами. Реализация интерфейсов в высокоуровневых языках программирования (через implements, duck typing др.). Полиморфизм. Переопределение и перегрузка методов. Позднее связывание (late binding). Использование интерфейсов для полиморфизма. Практика и архитектура: примеры иерархий классов, использование ООП в MVC-архитектуре; принципы SOLID (особенно LSP и ISP); применение в фреймворках.

Тема 9. ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ РАБОТЫ ВЕБ-ПРИЛОЖЕНИЯ ЗА СЧЁТ ХРАНЕНИЯ ДАННЫХ НА КЛИЕНТСКОЙ СТОРОНЕ. СЕССИИ, КУКИ И ХРАНЕНИЕ ДАННЫХ

Роль клиентского хранения в производительности веб-приложений: снижение нагрузки на сервер; ускорение загрузки страниц; повышение

отзывчивости интерфейса; улучшение пользовательского опыта (UX). Cookies (куки). Структура и формат (ключ-значение). Установка и чтение на клиенте и сервере. Срок действия, флаги безопасности (HttpOnly, Secure, SameSite). Применение: авторизация, персонализация, аналитика. Ограничения: объём (до 4 КБ), передача с каждым HTTP-запросом. Сессии. Понятие сессии и идентификатора. Хранение данных на сервере, идентификация через cookie. Жизненный цикл сессии. Примеры использования: корзина, авторизация, временные данные. Отличие от cookies: безопасность, объём, доступность. Web Storage API. localStorage и sessionStorage: различия и примеры. IndexedDB и другие клиентские базы данных (БД): возможности и ограничения; применение в офлайн-приложениях и PWA; Примеры: кэширование данных, хранение больших структур. Кэширование на клиентской стороне. HTTP-заголовки: Cache-Control, ETag, Expires, Last-Modified. Кэш браузера: HTML, CSS, JS, изображения. Service Workers и офлайн-доступ. Примеры: ускорение повторных загрузок, снижение трафика. Безопасность и ограничения. Уязвимости: XSS, CSRF, утечка данных. Шифрование и защита чувствительной информации. Политики безопасности – CORS, CSP. Практические сценарии: хранение корзины покупок в localStorage, использование cookies для запоминания языка интерфейса; сессии для авторизации и защиты маршрутов.

Раздел 3. ПРИЁМЫ, СРЕДСТВА И ИНСТРУМЕНТЫ ДЛЯ РАЗРАБОТКИ КЛИЕНТ-СЕРВЕРНЫХ ВЕБ-ПРИЛОЖЕНИЙ С ИСТОЧНИКОМ ДАННЫХ

Тема 10. ОРГАНИЗАЦИЯ МОДУЛЬНОГО ПОДХОДА ПРИ РАЗРАБОТКЕ ПРОЕКТОВ. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА СТОРОНЕ СЕРВЕРА

Понятие серверной логики и отличие от клиентской. Архитектура клиент-сервер и жизненный цикл HTTP-запроса. Обработка форм, маршрутизация, работа с базами данных. Принципы модульного подхода. Примеры модульной архитектуры (MVC, HMVC, микросервисы). Организация структуры проекта. Разделение на модули: контроллеры, модели, представления, сервисы. Использование пространств имён и автозагрузки. Инкапсуляция и интерфейсы между модулями. Использование интерфейсов и абстракций. Контейнеры зависимостей и внедрение зависимостей (DI). Примеры: сервисы авторизации, логирования, работы с БД. Работа с роутингом и контроллерами. Регистрация маршрутов по модулям. Обработка запросов и передача данных между слоями. RESTful-подход и модульные API. Безопасность серверной части: валидация, защита от SQL-инъекций.

Тема 11. ОРГАНИЗАЦИЯ РАБОТЫ С ФАЙЛАМИ И ФАЙЛОВОЙ СИСТЕМОЙ. РАБОТА С XML/JSON И ОБМЕН ДАННЫМИ. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СЦЕНАРИЕВ НА ОСНОВЕ ТЕХНОЛОГИИ AJAX

Понятие файла и файловой системы. Абсолютные и относительные пути. Операции с файлами: создание, чтение, запись, удаление. Работа с директориями: создание, обход, удаление. Поточковый ввод/вывод и буферизация. Обработка ошибок при работе с файлами. Права доступа и

безопасность. Структура XML-документа: элементы, атрибуты, схемы (DTD, XSD). Структура JSON: пары ключ-значение, вложенность, массивы. Сравнение XML и JSON: читаемость, гибкость, производительность. Парсинг и генерация XML/JSON. Примеры сериализации и десериализации данных. Обмен данными между клиентом и сервером: форматы передачи (JSON, XML, URL-кодировка); HTTP-запросы (методы GET, POST, PUT, DELETE); заголовки, коды состояния, MIME-типы. Технология Ajax. Понятие асинхронного взаимодействия. Принцип работы Ajax: запрос → сервер → ответ → обновление DOM. Использование XMLHttpRequest и fetch API. Обработка событий и обновление интерфейса без перезагрузки. Примеры: автозаполнение, фильтрация, динамическая подгрузка.

Тема 12. РАБОТА С БАЗАМИ ДАННЫХ. БЕЗОПАСНОСТЬ ВЕБ-ПРИЛОЖЕНИЙ

Реляционные и нереляционные БД (SQL vs NoSQL). Работа с БД на стороне сервера. Подключение к БД (в зависимости от языка программирования, например, PHP: PDO, mysqli; JS: Node.js + ORM). Подготовленные выражения и защита от SQL-инъекций. Обработка ошибок при работе с БД. Безопасность веб-приложений: общие принципы (принцип наименьших привилегий, аутентификация и авторизация, HTTPS и защита трафика, логирование и аудит). Работа с сессиями и куки. Механизмы хранения состояния пользователя. Безопасность сессий: HttpOnly, Secure, SameSite. Хранение токенов и защита от перехвата. Практика: безопасное CRUD-приложение. Создание формы с валидацией. Обработка данных с защитой от атак. Хранение и извлечение данных из БД. Логирование действий пользователя.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**ЛИТЕРАТУРА****ОСНОВНАЯ**

1. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – 5-е изд. – Санкт-Петербург : Питер, 2022. – 815 с.
2. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – Санкт-Петербург : Питер, 2021. – 816 с.
3. Кириченко, А. В. Web на практике. CSS, HTML, JavaScript, MySQL, PHP для fullstack-разработчиков / А. В. Кириченко, А. П. Никольский, Е. В. Дубовик. – Санкт-Петербург : Наука и Техника, 2021. – 432 с.
4. Маклафлин, Б. PHP и MySQL : исчерпывающее руководство / Б. Маклафлин. – 2-е изд. – Санкт-Петербург : Питер, 2017. – 544 с.
5. Скляр, Д. PHP. Рецепты программирования / Д. Скляр, А. Трахтенберг. – 3-е изд. – Санкт-Петербург : Питер, 2015. – 784 с.
6. Кузнецов, М. В. Объектно-ориентированное программирование на PHP / М. В. Кузнецов, И. В. Симдянов. – Санкт-Петербург : БХВ-Петербург, 2007. – 608 с.
7. Хавербеке, М. Выразительный JavaScript : современное веб-программирование / М. Хавербеке. – 3-е изд. – Санкт-Петербург : Питер, 2022. – 480 с.
8. Дронов, В. А. React 17. Разработка веб-приложений на JavaScript / В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2022. – 384 с.
9. Фрисби, М. JavaScript для профессиональных веб-разработчиков / М. Фрисби. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 1168 с.
10. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Санкт-Петербург : Питер, 2019. – 640 с.
11. Фримен, Э. Изучаем программирование на JavaScript / Э. Фримен, Э. Робсон. – Санкт-Петербург : Питер, 2018. – 640 с.
12. Новиков, В. А. Web-программирование : учебное пособие / В. А. Новиков. – Минск : Адукацыя і выхаванне, 2024. – 351 с.
13. Гарнаев, А. Ю. Web-программирование на Java и JavaScript / А. Ю. Гарнаев, С. Ю. Гарнаев. – Санкт-Петербург : БХВ-Петербург, 2002. – 1040 с.
14. Разработка дизайна веб-приложений : учебник : для студентов учреждений среднего профессионального образования, обучающихся по специальности «Информационные системы и программирование» / Т. В. Мусаева, Е. В. Поколодина, М. А. Трифанов, Е. С. Хайбрахманова. – 3-е изд., стер. – Москва : Академия, 2023. – 256 с.
15. Форсье, Д. Django. Разработка веб-приложений на Python / Д. Форсье, П. Биссекс, У. Чан. – Санкт-Петербург : Символ, 2013. – 456 с.
16. Тузовский, А. Ф. Проектирование и разработка Web-приложений : учебное пособие / А. Ф. Тузовский. – Москва : Юрайт, 2021. – 218 с.
17. Разработка web-приложений для мобильных систем : пособие / А. О. Рак, А. В. Михалькевич, А. С. Середа [и др.]. – Минск : БГУИР, 2019. – 130 с.

18. Прайс, М. Д. С# 10 и .NET 6. Современная кросс-платформенная разработка / М. Д. Прайс. – 6-е изд. – Санкт-Петербург : Питер, 2023. – 848 с.
19. Прайс, М. Д. С# 9 и .NET 5 : разработка и оптимизация / М. Д. Прайс. – 5-е изд. – Санкт-Петербург : Питер, 2022. – 832 с.
20. Стиллмен, Э. Изучаем С# / Э. Стиллмен, Д. Грин. – 4-е изд. – Санкт-Петербург : Питер, 2022. – 768 с.
21. Арораа, Г. Паттерны проектирования для С# и платформы .NET Core / Г. Арораа, Д. Чилберто. – Санкт-Петербург : Питер, 2021. – 352 с.
22. Албахари, Д. С# 9.0. Справочник : полное описание языка / Д. Албахари ; пер. с англ. – Москва ; Санкт-Петербург : Диалектика, 2021. – 1056 с.
23. Янг, А. Node.js в действии / А. Янг, Б. Мек, М. Кантелон. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 432 с.
24. Постолиит, А. Основы искусственного интеллекта в примерах на Python : самоучитель / А. Постолиит. – 2-е изд. – Санкт-Петербург : БХВ-Петербург, 2024. – 448 с.
25. Левашов, П. Python с нуля / П. Левашов. – Санкт-Петербург : Питер, 2024. – 448 с.
26. Python и DevOps : ключ к автоматизации Linux / Н. Гифт, К. Берман, А. Деза, Г. Георгиу. – Санкт-Петербург : Питер, 2024. – 544 с.
27. Ван, Хорн II Брюс М. PyCharm: профессиональная работа на Python / Ван Хорн II Брюс М., К. Нгуен ; [пер. с англ. И. Л. Люско]. – Москва : ДМК Пресс, 2024. – 618 с.
28. Персиваль, Г. Паттерны разработки на Python : TDD, DDD и событийно-ориентированная архитектура / Г. Персиваль, Б. Грегори. – Санкт-Петербург : Питер, 2023. – 336 с.
29. Бизли, Д. Python. Исчерпывающее руководство / Д. Бизли. – Санкт-Петербург : Питер, 2023. – 368 с.
30. Плас, Д. В. Python для сложных задач : наука о данных и машинное обучение / Д. В. Плас. – Санкт-Петербург : Питер, 2023. – 576 с.
31. Прохорёнок, Н. А. Python 3 и PyQt 6. Разработка приложений / Н. А. Прохорёнок, В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2023. – 832 с.
32. Фаулер, М. Asyncio и конкурентное программирование на Python / М. Фаулер. – Москва : ДМК Пресс, 2023. – 398 с.
33. Доусон, М. Програмируем на Python / М. Доусон. – [3-е изд.]. – Санкт-Петербург : Питер, 2022. – 416 с.
34. Цукалос, М. Golang для профи : создаём профессиональные утилиты, параллельные серверы и сервисы / М. Цукалос. – 3-е изд. – Санкт-Петербург : Питер, 2024. – 624 с.
35. Индрасири, К. gRPC: запуск и эксплуатация облачных приложений. Go и Java для Docker и Kubernetes / К. Индрасири, Д. Куруппу. – Санкт-Петербург : Питер, 2021. – 224 с.
36. Казанский, А. А. Разработка приложений на Swift и SwiftUI с нуля / А. А. Казанский. – 2-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2022. – 416 с.

37. Усов, В. Swift. Основы разработки приложений под iOS, iPadOS и macOS / В. Усов. – 6-е изд. – Санкт-Петербург : Питер, 2022. – 544 с.
38. Лой, М. Программируем на Java / М. Лой, П. Нимайер, Д. Лук. – 5-е изд. – Санкт-Петербург: Питер, 2023. – 544 с.
39. Макгрегор, Д. От Java к Kotlin / Д. Макгрегор, Н. Прайс ; пер. с англ. – Санкт-Петербург : БХВ-Петербург, 2023. – 448 с.
40. Шилдт, Г. Полное руководство. Java / Г. Шилдт. – 12-е изд. – Москва : Диалектика, 2022. – 1344 с.
41. Парлог, Н. Система модулей Java / Н. Парлог ; пер. с англ. А. Павлова. – Санкт-Петербург : Питер, 2021. – 464 с.
42. Смолякова, О. Г. Основы разработки веб-приложений на языке программирования Java : учебно-методическое пособие / О. Г. Смолякова. – Минск : БГУИР, 2019. – 131 с.
43. Java Persistence API и Hibernate / К. Бауэр, Г. Кинг, Г. Грегори ; пер. с англ. Д. А. Зинкевича. – Москва : ДМК Пресс, 2017. – 632 с.
44. Эккель, Б. Философия Java / Б. Эккель. – 4-е полное изд. – Санкт-Петербург : Питер, 2018. – 1168 с.
45. Лафоре, Р. Структуры данных и алгоритмы Java / Р. Лафоре. – 2-е изд. – Санкт-Петербург : Питер, 2018. – 704 с.
46. Седжвик, Р. Computer Science : основы программирования на Java, ООП, алгоритмы и структуры данных / Р. Седжвик. – Санкт-Петербург : Питер, 2018. – 1072 с.
47. Гонсалвес, Э. Изучаем Java EE 7 / Э. Гонсалвес. – Санкт-Петербург : Питер, 2017. – 640 с.
48. Курняван, Б. Программирование web-приложений на языке Java / Б. Курняван. – Москва : Лори, 2014. – 880 с.
49. Васильев, А. Н. Java. Объектно-ориентированное программирование : для магистров и бакалавров / А. Н. Васильев. – Санкт-Петербург : Питер, 2014. – 400 с.
50. Гупта, А. Java EE 7. Основы / А. Гупта ; пер. с англ. А. Г. Гузикевича. – Москва : Вильямс, 2014. – 336 с.
51. Герман, О. В. JAVA и Интернет-бизнес / О. В. Герман, Ю. О. Герман. – Минск : Бестпринт, 2010. – 384 с.
52. Holzner, S. JAVA 2 : Programming ; Black Book / S. Holzner. – New Delhi : Paraglyph Press, 2010. – 1460 p.
53. Эккель, Б. Философия Java / Б. Эккель. – 4-е изд. – Санкт-Петербург : Питер, 2009. – 640 с.
54. Хемраджани, А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse / А. Хемраджани ; пер. с англ. – Москва : Вильямс, 2008. – 352 с.
55. Хорстманн, К. С. Java 2 : в 2 т. / К. С. Хорстманн, Г. Корнелл. – 8-е изд. – Москва : Вильямс, 2008. – Т. 1 Основы. – 816 с.
56. Хорстманн, К. С. Java 2 : в 2 т. / К. С. Хорстманн, Г. Корнелл. – 8-е изд. – Москва : Вильямс, 2009. – о Т. 2 : Тонкости программирования. – 992 с.
57. Герман, О. В. Программирование на Java и C# для студента / О. В. Герман, Ю. О. Герман. – Санкт-Петербург : БХВ-Петербург, 2005. – 512 с.

58. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML : учебное пособие / Э. Г. Галиаскаров, А. С. Воробьев. – Москва : Юрайт, 2022. – 125 с.
59. Олифер, В. Г. Компьютерные сети : принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – Санкт-Петербург : Питер, 2021. – 1008 с.
60. Букатов, А. А. Компьютерные сети : расширенный начальный курс : учебное пособие / А. А. Букатов, С. А. Гуда. – Санкт-Петербург : Питер, 2020. – 496 с.
61. Умрихин, Е. Разработка веб-приложений с помощью ASP.Net Core MVC / Е. Умрихин. – Санкт-Петербург : БХВ-Петербург, 2023. – 416 с.
62. Тепляков, С. Паттерны проектирования на платформе .NET / С. Тепляков. – Санкт-Петербург : Питер, 2021. – 320 с.
63. Рихтер, Д. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Д. Рихтер. – 4-е изд. – Санкт-Петербург : Питер, 2019. – 896 с.
64. Троелсен, Э. Язык программирования C# 5.0 и платформа .NET 4.5 / Э. Троелсен. – 6-е изд. – Москва : Вильямс, 2016. – 1312 с.
65. .NET. Сетевое программирование / В. Кумар [и др.]. – Москва : Лори, 2014. – 400 с.
66. Перри, Б. У. Java сервлеты и JSP сборник рецептов / Б. У. Перри. – 2-е изд. – Москва : Кудиц-образ, 2006. – 768 с.
67. Мориссо-Леруа, Н. ORACLE8i Java-компонентное программирование при помощи EJB, COBRA и JSP / Н. Мориссо-Леруа. – Москва : Лори, 2002. – 484 с.
68. Фленов, М. Е. C# глазами хакера / М. Е. Фленов. – Санкт-Петербург : БХВ-Петербург, 2023. – 224 с.
69. Тузовский, А. Ф. Проектирование и разработка Web-приложений : учебное пособие / А. Ф. Тузовский. – Москва : Юрайт, 2021. – 218 с.
70. Сеттер, Р. В. Изучаем Java на примерах и задачах / Р. В. Сеттер. – Санкт-Петербург : Наука и техника, 2016. – 240 с.
71. Флэнаган, Д. Java в примерах : справочник / Д. Флэнаган. – Санкт-Петербург : Символ, 2016. – 664 с.
72. Нимейер, П. Программирование на Java / П. Нимейер, Д. Леук. – Москва : Эксмо, 2014. – 1312 с.
73. Портянкин, И. Swing. Эффективные пользовательские интерфейсы / И. Портянкин. – Москва : Лори, 2011. – 608 с.
74. Прохоренок, Н. JavaFX / Н. Прохоренок. – Санкт-Петербург : БХВ-Петербург, 2020. – 768 с.
75. Куликов, С. С. Работа с MySQL, MS SQL Server и Oracle в примерах : учебно-методическое пособие : в 2 ч. / С. С. Куликов, Е. Е. Фадеева. – Минск : БГУИР, 2019. – Ч. 1. – 287 с.
76. Садаладж, П. Д. NoSQL : новая методология разработки нереляционных баз данных / П. Д. Садаладж, М. Фаулер. – Москва : Вильямс, 2017. – 192 с.
77. Мак-Лахлин, Б. Java и XML / Б. Мак-Лахлин ; пер. с англ. – Санкт-Петербург : Символ, 2015. – 544 с.

ДОПОЛНИТЕЛЬНАЯ

78. Котеров, Д. В. PHP 8 / Д. В. Котеров, И. В. Симдянов ; Д. В. Котеров, И. В. Симдянов. – Санкт-Петербург : БХВ-Петербург, 2023. – 992 с.
79. Дронов, В. А. Laravel 9. Быстрая разработка веб-сайтов на PHP / В. А. Дронов ; В. А. Дронов. – Санкт-Петербург : БХВ-Петербург, 2023. – 768 с.
80. Татро, К. Создаем динамические веб-сайты на PHP / К. Татро, П. Макинтайр. – 4-е междунар. изд. – Санкт-Петербург : Питер, 2021. – 544 с.
81. Прохоренок, Н. А. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н. А. Прохоренок, В. А. Дронов. – 5-е изд., перераб. и доп. – Санкт-Петербург : БХВ-Петербург, 2019. – 912 с.
82. Хорстман, К. С. Современный JavaScript для нетерпеливых / К. С. Хорстман ; К. С. Хорстман. – пер. с англ. А. А. Слинкина. – Москва : ДМК Пресс, 2023. – 288 с.
83. Скотт, А. Д. JavaScript. Рецепты для разработчиков / А. Д. Скотт, М. Макдоналд, Ш. Пауэрс. – 3-е изд. – Санкт-Петербург: Питер, 2023. – 528 с.
84. Свекис, Л. JavaScript с нуля до профи. / Л. Свекис, М. ван Путтен, Р. Персиваль. – Санкт-Петербург : Питер, 2023. – 480 с.
85. Лафоре, Р. Структуры данных и алгоритмы в Java / Р. Лафоре. – 2-е изд. – Санкт-Петербург : Питер, 2015. – 704 с.
86. Блох, Д. Java. Эффективное программирование / Д. Блох. – Москва : Лори, 2016. – 464 с.
87. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин. – Санкт-Петербург : Питер, 2015. – 464 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

контролируемая самостоятельная работа в виде решения индивидуальных заданий в аудитории во время проведения лабораторных занятий под контролем преподавателя;

работа с основной и дополнительной литературой, интернет-ресурсами.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ СТУДЕНТОВ

Примерным учебным планом по специальности 6-05-0611-01 «Информационные системы и технологии» в качестве формы промежуточной аттестации по учебной дисциплине «Высокоуровневые языки программирования» рекомендуется зачет. Оценка учебных достижений студентов производится по системе «зачтено/не зачтено».

Для текущего контроля по учебной дисциплине и диагностики компетенций могут использоваться следующие формы:

отчеты по лабораторным работам с их устной защитой;

защита индивидуальной практической работы;
 защита контрольной работы;
 устный опрос;
 электронные тесты;
 коллоквиум.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

проблемное обучение (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемое на лекционных занятиях;
 устное изложение учебного материала на лекциях, сопровождаемое показом и демонстраций слайдов, видеоматериалов;
 учебно-исследовательская деятельность, творческий подход, реализуемые на лабораторных занятиях.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Синтаксис и базовые конструкции языка. Управляющие конструкции и циклы. Работа с массивами и коллекциями.
2. Функции и принципы модульности.
3. Основы объектно-ориентированного программирования.
4. Работа с формами и обработка данных.
5. Работа с файлами и сериализация данных. Хранение данных на клиентской и/или серверной стороне.
6. Асинхронный обмен данными с использованием Ajax. Варианты оптимизации проектных программных решений.
7. Работа с базами данных. Разработка модульного веб-приложения.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ

(необходимого оборудования, наглядных пособий и т. п.)

1. Операционная система Microsoft Windows 7 и выше.
2. Комплект разработчика приложений JDK версии 8 и выше.
3. Комплект разработчика приложений SDK.
4. Интегрированная среда разработки на объектно-ориентированном языке программирования IntelliJ IDEA или аналоги.
5. Интегрированная среда разработки PHPStorm.
6. phpDesigner – среда обработки, обладающая широким функционалом.
7. XAMPP – дистрибутив Apache, который помимо PHP содержит также Perl и MariaDB.
8. Интегрированная среда разработки на объектно-ориентированном языке программирования MS Visual Studio или аналоги.
9. Система управления базами данных MySQL или аналоги.

10. Браузер Google Chrome или аналоги.
11. CASE-инструмент для проектирования и конструирования программного обеспечения Enterprise Architect 7.1 и выше или аналоги.
12. Интегрированная среда разработки Visual Studio Code или редакторы: Sublime Text, Atom, WebStorm, notepad++.
13. Программная платформа Node.js.
14. Web-сервера Tomcat, JBoss and Glassfish.
15. Инструменты для сборки приложений: Gradle, Maven и Ant.
16. JaCoCo – инструмент для оценки охвата кода юнит тестами.
17. JMeter – инструмент с открытым исходным кодом для тестирования производительности приложений.
18. JUnit – библиотека для модульного тестирования программного обеспечения на Java.
19. Spark – веб-фреймворк для Java, предназначенный для быстрой разработки с минимальными накладными расходами.
20. Фреймворк Spring – инструмент для инъекций зависимостей.