

# БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**УТВЕРЖДАЮ**

Проректор по учебной работе и  
образовательным инновациям

О.Г. Прохоренко

«05» июля 2023 г.

Регистрационный № УД – 461/б.

## **ВЕБ-ПРОГРАММИРОВАНИЕ**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности:**

**6-05-0533-07 Математика и компьютерные науки**

**Профилизации:**

**Математическое и программное обеспечение мобильных устройств**

**Веб- программирование и интернет-технологии**

2023 г.

Учебная программа составлена на основе ОСВО 6-05-0533-07-2023, примерного учебного плана по специальности 6-05-0533-07 «Математика и компьютерные науки», регистрационный № 6-05-05-028/пр. от 31.01.2023 г, учебных планов БГУ: № 6-5.4-55/02 от 15.05.2023 г. и № 6-5.4-55/03 от 15.05.2023 г., учебных планов № 6-5.4-55/22з. от 31.05.2023 г. и № 6-5.4-55/23з. от 31.05.2023 г., учебных планов № 6-5.4-55/12и. от 31.05.2023 г., № 6-5.4-55/13и. от 31.05.2023 г.

**СОСТАВИТЕЛИ:**

**Сергей Александрович Барвенков**, доцент кафедры веб-технологий и компьютерного моделирования Белорусского государственного университета, кандидат физико-математических наук, доцент;

**РЕЦЕНЗЕНТЫ:**

**Н.Г. Токаревская**, доцент кафедры информационных технологий УО "Белорусский государственный экономический университет", кандидат физико-математических наук, доцент.

**А.С. Кравчук**, доцент кафедры экономической информатики УО "Белорусский государственный экономический университет", кандидат физико-математических наук, доцент.

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

Кафедрой веб-технологий и компьютерного моделирования БГУ  
(протокол № 11 от 24.05.2023 г.)

Научно-методическим советом БГУ  
(протокол № 9 от 29.06.2023 г.)

Заведующий кафедрой \_\_\_\_\_



М.В. Игнатенко

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время создание web-приложений является одним из наиболее интенсивно развивающихся разделов информационных технологий. Это связано как с бурным развитием интернета, так и с желанием практически любой организации во всех сферах жизнедеятельности человека иметь представительство в сети интернет не только в виде статического сайта, но и полноценного приложения для взаимодействия с пользователями. Все это определяет важность учебной дисциплины «*Веб-программирование*» в учебном процессе, а также обуславливает необходимость внесения своевременных изменений и дополнений в ее содержание.

*Центральной идеей* образования по дисциплине «*Веб-программирование*» является необходимость обучения студентов современным подходам и методам решения прикладных задач создания и поддержки динамических сайтов, а также принципами их грамотного построения.

*Второй важнейшей идеей* обучения является подготовка студентов к практической работе в области программирования для веб.

### **Цели и задачи учебной дисциплины**

Дисциплина «*Веб-программирование*» имеет прикладную направленность.

*Основными целями* дисциплины являются:

- дать теоретическую подготовку по базовым веб-технологиям и по созданию веб-приложений;
- дать представление о тенденциях и направлениях в развитии современных информационных технологий, путей и задач развития компьютерных сетей и Интернет;
- дать практические навыки использования сервисов сети internet и основных протоколов;
- дать практические навыки использования языка HTML5, спецификации CSS3 и принципов вёрстки при создании динамических веб-страниц;
- дать теоретические знания языка JavaScript;
- познакомить с серверными технологиями;
- дать практические навыки создания веб-приложений.

*Основными задачами* дисциплины являются:

- получить базовые знания по представлению, организации и передаче информации и структуре Web;
- изучить этапы разработки веб-сайта;
- использовать популярные принципы верстки и дизайна сайтов;
- изучить современные библиотеки для создания веб-приложений;
- изучить возможности серверных технологий, которые позволяют реализовать распределенное приложение типа клиент-сервер;
- развития умения и навыков выбора современных методов создания веб-сайтов и веб-приложений.

Опыт преподавания дисциплины «*Веб-программирование*» на механико-математическом факультете БГУ показывает, что обучение на практических занятиях должно проводиться в двух направлениях: изучения основ веб-технологий и выполнения индивидуальных работ. При этом только непосредственное общение студента с конкретными задачами кроме возможности закрепить лекционный материал, помогает дать общее представление и выработать необходимую интуицию для нахождения эффективных путей решения задач верстки и создания UI.

При этом значительно возрастает роль самостоятельной работы студентов над предметом, без чего его успешное освоение представляется маловероятным. Общая оценка качества усвоения студентами учебного материала осуществляется в ходе выполнения индивидуальных заданий.

Каждый обучаемый получает задания по созданию отдельных веб-страниц, сайтов и частей серверных приложений.

**Место учебной дисциплины** в системе подготовки специалиста с высшим образованием.

Дисциплина «*Веб-программирование*» относится к модулю «Программирование» государственного компонента.

**Связи с другими учебными дисциплинами**, включая учебные дисциплины компонента учреждения высшего образования, дисциплины специализации и др.

Программа учебной дисциплины «*Веб-программирование*» составлена с учетом межпредметных связей и программ по смежным дисциплинам. Её изучение базируется на знаниях, полученных при изучении дисциплин модуля «Программирование» государственного компонента: «Методы программирования» и «Технологии программирования».

### **Требования к компетенциям**

Освоение учебной дисциплины должно обеспечить формирование у студентов следующих компетенций:

#### **универсальные компетенции:**

– УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий.

– УК-5. Быть способным к саморазвитию и совершенствованию в профессиональной деятельности.

#### **базовые профессиональные компетенции:**

– БПК-3 Применять теоретические знания и навыки в самостоятельной исследовательской деятельности.

– БПК-6 Применять современные технологии и базовые конструкции языков программирования для реализации алгоритмических прикладных задач и разработки веб-проектов.

#### **специализированные компетенции:**

– СК-1. Осуществлять анализ контекста и поставленной проблемы, ар-

гументированно выбирать оптимальный способ ее решения, согласовывать частичные проекты решения в общую согласованную архитектуру, выполнять реализацию проекта с учетом оценки накопленных и поступающих данных.

В результате освоения учебной дисциплины студент должен:

***знать:***

- основные понятия, терминологию и структуру компьютерных сетей и интернет;
- использование HTML5, CSS3 и библиотек для адаптивной верстки веб-страниц;
- основы JavaScript;
- основы создания web-приложений с использованием популярных библиотек;
- основы программирования серверной части веб-приложений;

***уметь:***

- проектировать интерфейс веб-приложений;
- верстать веб-страницы, создавать и использовать CSS на этих страницах;
- создавать динамические страницы средствами JavaScript;
- создавать серверную часть сайта;
- программировать обработку на сервере данных, которые вводит пользователь на клиентской части веб-приложения;

***владеть:***

- основными приемами проектирования веб-приложений;
- инструментами разработки дизайна сайтов;
- навыками работы в “инструментах разработчика” основных браузеров;
- средствами разработки клиентских и серверных частей сайта.

### **Структура учебной дисциплины**

Дисциплина изучается во 2,3,4 семестрах очной формы получения высшего образования и в 3-6 семестрах заочной формы получения высшего образования.

Всего на изучение учебной дисциплины «Веб-программирование» отведено:

– в очной форме получения высшего образования: 268 часов, в том числе 104 аудиторных часов, из них: лекции – 50 часов, лабораторные занятия – 44 часа, управляемая самостоятельная работа – 10 часов.

– в заочной форме получения высшего образования: 268 часов, в том числе 24 аудиторных часов, из них: лекции – 12 часов, лабораторные занятия – 12 часов.

Трудоемкость учебной дисциплины составляет 8 зачетных единиц.

Форма промежуточной аттестации по учебной дисциплине – зачет в 2, 3 семестре, экзамен в 4 семестре очной формы обучения, зачет в 4, 5 семестре, экзамен в 6 семестре заочной формы обучения.

### Очная форма обучения

Курс, семестры	Экзамен семестр	Зачет, семестр	Всего часов	В том числе аудиторных	Из них		
					Лекции	Лабораторные занятия	Количество часов УСП
1 к,2 с		2	90	34	16	16	2
2 к,3 с		3	88	36	18	14	4
2 к,4 с	4		90	34	16	14	4
Всего			268	104	50	44	10

### Заочная форма обучения

Семестры	Экзамен семестр	Зачет, семестр	Контрольная работа, семестр	Всего часов	В том числе аудиторных	Из них	
						Лекции	Лабораторные занятия
2 к,3 с				60	4	2	2
2 к,4 с		4		60	4	2	2
3 к,5 с		5	5	74	8	4	4
3 к,6 с	6		6	74	8	4	4
Всего				268	24	12	12

Планируется проведение занятий с использованием онлайн-консультаций, компьютеров в аудитории и видеопроектора. Для очной формы обучения после 2 и 3 семестров сдаются зачёты по результатам выполненных заданий, после 4 семестра – экзамен. Для заочной формы обучения после 4 и 5 семестров сдаются зачёты по результатам выполненных заданий, после 6 семестра – экзамен.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### **Тема 1. Введение.**

Цель, содержание и особенности учебной дисциплины. История Интернета. Технологии и протоколы, обеспечивающие работу глобальной сети. Протоколы TCP/IP. IP – адресация. DNS – адресация. Протокол http. Хостинг.

Виды Web – приложений (сервисы интернета). Браузеры. Общие алгоритмы и принципы отображения страниц. Режим разработчика в браузере.

Web – дизайн и проектирование. Этапы создания сайта. Современные подходы к промышленному созданию веб-сервисов и приложений. Язык программирования программных и информационных систем UML.

### **Тема 2. Язык JavaScript. Создание динамических приложений-клиентов.**

Типы данных. Переменные. Функции. Операторы. Массивы. Объекты. Работа с DOM.

События. Делегирование.

Модули в JavaScript. Веб-компоненты

Событийный цикл: микрозадачи и макрозадачи. Промисы, async/await

Минификация и сборка проекта.

### **Тема 3. Использование библиотек для создания клиентских приложений.**

Библиотека React.

Сборка проекта. Размещение проекта в сети.

Паттерн Flux. Библиотека Redux.

### **Тема 4. Создание серверных приложений.**

Клиент-серверные технологии. Протокол HTTP.

Web-серверы Apache, nginx. Администрирование web-серверов.

Взаимодействие с клиентом, передача параметров, методы.

JavaScript на сервере. Технологии NodeJS.

Язык PHP. Классы и объекты.

CMS и фреймворки. Веб-фреймворк Laravel.

СУБД MySQL и MSSql Server и работа с приложениями на сервере. MongoDB, как пример нереляционных БД.

## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования

Номер темы	Название темы	Количество аудиторных часов					Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР		
1	2	3	4	5	6	7	8	9
<b>2 семестр (34 часа)</b>								
1	Введение	2					[1, 5, 6]	Собеседование
2	Язык JavaScript. Создание динамических приложений-клиентов.	16			16	2	[1,2,6]	Опрос, защита практических заданий, электронные контрольные тесты
<b>3 семестр (36 часов)</b>								<b>зачет</b>
3	Использование библиотек для создания клиентских приложений	18			14	4	[1,2,6]	Опрос, отчет по лабораторным работам, защита практических заданий, электронные контрольные тесты
<b>4 семестр (34 часа)</b>								<b>зачет</b>
4	Создание серверных приложений	16			14	4	[1,3-7]	Опрос, защита практических заданий, отчет по лабораторным работам, электронные контрольные тесты
<b>Всего</b>		<b>50</b>			<b>44</b>	<b>10</b>		<b>Экзамен</b>



## УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Заочная форма получения высшего образования

Номер темы	Название темы	Количество аудиторных часов					Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР		
1	2	3	4	5	6	7	8	9
1	Язык JavaScript. Создание динамических приложений-клиентов.	4			4		[1,2,5, 6]	Опрос, защита практических заданий, электронные контрольные тесты
2	Использование библиотек для создания клиентских приложений	4			4		[1,2,6]	Опрос, защита практических заданий
3	Создание серверных приложений	4			4		[1,3-7]	Опрос, отчет по лабораторным работам
<b>Всего</b>		<b>12</b>			<b>12</b>			

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Перечень основной литературы

1. Романчик, В. С. Веб-программирование : учеб. пособие для студ. учреждений высш. образования по математическим спец. / В. С. Романчик ; БГУ. - Минск : БГУ, 2013. - 407 с. - URL: <http://elib.bsu.by/handle/123456789/90791>.
2. Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 : учебное пособие / А. В. Диков. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2023. - 184 с. - URL: <https://reader.lanbook.com/book/318443#1>.
3. Дронов В. А. JavaScript: 20 уроков для начинающих. — (Для начинающих) / В.А. Дронов. - Санкт-Петербург : БХВ-Петербург, 2020. - 352 с. - URL: <https://ibooks.ru/bookshelf/385776/reading>.
4. Диков, А. В. Клиентские технологии веб-программирования: JavaScript и DOM : учебное пособие / А. В. Диков. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2020. - 121 с. - URL: <https://reader.lanbook.com/book/126934>.
5. Заяц, А. М. Проектирование и разработка web-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие / А. М. Заяц, Н. П. Васильев. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2019. - 118 с.
6. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML 5 / Робин Никсон ; [пер. с англ. С. Черников]. - 6-е изд. ; включает React, PHP 8 & MySQL 8 . - Санкт-Петербург ; Москва ; Минск : Питер, 2023. - 830 с. - URL: <https://ibooks.ru/bookshelf/386792/reading>.
7. Дронов, В. А. React 17. Разработка веб-приложений на JavaScript / Владимир Дронов. - Санкт-Петербург : БХВ-Петербург, 2022. - 384 с. - URL: <https://ibooks.ru/bookshelf/385763/reading>.

### Перечень дополнительной литературы

1. Хрусталеv А. HTML5 + CSS3. Основы современного WEB-дизайна / А. Хрусталеv, А. Кириченко. – СПб: Символ-Плюс, 2018. – 346 с.
2. Грант К. CSS для профи/ Кит Грант. – СПб: Питер, 2021. - 496 с.
3. Фрэйн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств Пер. с англ., СПб: Питер, 2017. – 272 с.
4. Дронов, В. HTML5, CSS3 и Web 2.0. Разработка современных Web-сайтов / Владимир Дронов. - М.: БХВ-Петербург, 2016. - 416 с.
5. Мержевич, В. HTML и CSS на примерах / Влад Мержевич. - М.: "БХВ-Петербург", 2016. - 448 с.
6. Робсон, Э. Изучаем HTML, XHTML и CSS / Элизабет Робсон, Эрик Фримен ; [пер. с англ. В. Черник]. - 2-е изд. - Москва ; Санкт-Петербург ; Минск : Питер, 2022. - 720 с. - URL: <https://ibooks.ru/bookshelf/377028/reading>.

7. Янцев, В. В. JavaScript. Картинки, галереи, слайдеры : учебное пособие / В. В. Янцев. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2022. - 250 с. - URL: <https://reader.lanbook.com/book/256067>.

8. Веллинг, Т. Разработка веб-приложений с помощью PHP и MySQL / Веллинг Томсон – М.: 2017. – 696с.

9. Мейер Э. CSS: Полный справочник/ Эрик Мейер, Эстелл Уэйл. – Пер. с англ. – 4-е издание, СПб: Символ-Плюс, 2019. – 576 с.

10. Вейл Э."HTML5. Разработка приложений для мобильных устройств" Питер, 2015 год, 480 стр.

11. Татро, К. Создаем динамические веб-сайты на PHP = Programming PHP. Creating Dynamic Web Pages / Кевин Татро, Питер Макинтайр ; [пер. с англ. Е. Матвеева]. - 4-е междунар. изд. - Санкт-Петербург ; Москва ; Минск : Питер, 2021. - 541 с. - URL: <https://ibooks.ru/bookshelf/376967/reading>.

12. Грофф Д.Энциклопедия SQL / Грофф Д., Вайнберг П.. – М.: 2003. – 896с.

13. [htmlbook.ru](http://htmlbook.ru) | Для тех, кто делает сайты ([htmlbook.ru/](http://htmlbook.ru/))

14. Современный учебник JavaScript ([learn.javascript.ru/](http://learn.javascript.ru/))

15. JavaScript Courses & Tutorials ([www.codecademy.com/learn/javascript](http://www.codecademy.com/learn/javascript))

## **Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки**

Рекомендуются следующие формы диагностики компетенций.

### **Устная форма**

1. Опрос, собеседование

### **Письменная форма**

1. Опрос.
2. Электронные контрольные тесты.

### **Устно-письменная форма**

1. Отчеты по лабораторным работам с их устной защитой.
2. Защита практических заданий.
3. Экзамен, зачет.

Оценка за ответы на лекциях (опрос) и лабораторных занятиях может включать в себя полноту ответа, наличие аргументов, примеров из практики и т.д.

При оценке лабораторных работ студента необходимо обращать внимание на: актуальность используемых методов верстки страниц проекта, содержание и степень развития проекта, оригинальность дизайна сайта, наличие самооценки студентом собственных работ (рефлексивные замечания, реплики, комментарии, проекты изменений).

Формой промежуточной аттестации по дисциплине «Веб-программирование» учебным планом предусмотрен **зачет и экзамен**.

Контроль управляемой самостоятельной работы (УСР) проводится преподавателем, как правило, во время аудиторных занятий и осуществляется в виде:

- опроса на лекциях и аудиторных занятиях;
- защиты практических заданий.

При формировании итоговой отметки используется рейтинговая система оценки знаний обучающихся, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов в ходе проведения контрольных мероприятий текущей аттестации.

Примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации:

Формирование отметки за текущую аттестацию:

- опрос – 10 %;
- отчеты по лабораторным работам – 60 %;
- электронные контрольные тесты – 30%.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей аттестации и экзаменационной отметки с учетом их весовых коэффици-

циентов Вес отметки по текущей аттестации составляет 40 %, экзаменационной отметки – 60 %.

### **Примерный перечень заданий для управляемой самостоятельной работы студентов**

#### **Тема 2. Язык JavaScript. Создание динамических приложений-клиентов. (2ч)**

Студентам предлагается выполнить одно из следующих заданий:

1. Создание Web-приложения по теме. Студентам необходимо разработать проект сайта и создать SPA сайт или многостраничный сайт, используя JavaScript, запросы на сервер, современные подходы к верстке. Предлагаемые для разработки темы приложений:
  - компьютерный клавиатурный тренажер
  - новостной сайт с подключаемыми с других сайтов информерами (курсы валют, прогноз погоды, спортивные новости, анекдоты и т.п.).
  - Интернет-галерея фотографий с возможностью оценивания. Очередность отображения фотографий зависит от рейтинга.
  - Интернет-магазин (по продаже цветов, по продаже компьютерной техники, по продаже настольных игр и др.).
2. Создание интерактивной анимации, визуализирующей несколько простейших физических процессов (бросок тела под углом к горизонту, отскок мяча от плоскости, магнитное поле и т.п.).
3. Создание страницы, визуализирующей построение треугольника по трем элементам которые интерактивно задаются (3 стороны, 2 стороны и угол, 2 угла и сторона, высота и 2 стороны и т.п.).

**Форма контроля** - защита практических заданий

#### **Тема 3. Использование библиотек для создания клиентских приложений (4ч).**

Студентам предлагается выполнить одно из следующих заданий:

1. Реализовать свой блог, используя Next.js. Блог – это набор предсгенерированных HTML страниц. Блог доступен онлайн, для этого можно использовать, например, <https://vercel.com/> или <https://www.netlify.com/>; HTML страницы проще всего генерировать из markdown файлов. Библиотеку для парсинга и конвертации .md в HTML предлагается выбрать самостоятельно.
2. Реализовать TODO List используя SolidJS <https://www.solidjs.com/>  
Требования: Приложение позволяет создать, удалить и редактировать todo; Todo не сбрасываются после перезагрузки страницы.
3. Выполнить задание по использованию Redux в TodoList или Notion, используя Redux-Toolkit

**Форма контроля** - защита практических заданий.

#### **Тема 4. Создание серверных приложений. (4ч)**

Студентам предлагается выполнить одно из следующих заданий:

1. Проверить, является ли заданная целочисленная матрица латинским квадратом. Латинским квадратом порядка  $n$  называется квадратная таблица размером  $n \times n$ , каждая строка и каждый столбец которой содержат все числа от 1 до  $n$ .
2. Поздравления. По заданному в файле списку фамилий напечатать каждому упомянутому в списке поздравление к определенному празднику. Чтобы избежать шаблона, перечень же-лаемых благ выбирать как случайное подмножество из заготовленного списка (например, здоровья, счастья, продвижения по службе, долголетия и т. д.). Можно сделать переменными и название праздника – для универсальности программы.
3. Имеется список членов коллектива с указанием принадлежности каждого к различным общественным организациям (профком, ученый совет, общество книголюбов, секция футбола и т. д.). Напечатать приглашение всем членам на очередное заседание указанной организации. Задается только вид организации, место и время сбора.
4. Создать простую игру типа: “Угадай задуманное число”, “black jack”.
5. Составить список посетителей за текущий день (неделю, месяц). IP-адрес посетителя страницы находится как элемент суперглобального массива `$_SERVER['REMOTE_ADDR']`. Если посетители приходят из одной организации с проху-server, то IP-адрес у них будет одинаковый. В некоторых случаях можно получить адреса машин в подсети с помощью функции `HTTP_X_FORWARDED_FOR`.
6. Спроектировать и создать простую базу данных из 2-3 таблиц. Например, «Видеотека», «Сувениры», «Заказы», «Достопримечательности». Организовать средствами PHP соединение с БД, выполнить запрос на извлечение информации из БД, когда клиент в форме на html странице выбирает желаемые параметры. Результаты выполнения запроса передать клиенту и отобразить. Создать класс на добавление информации в БД.

**Форма контроля** - защита практических заданий.

#### **Описание инновационных подходов и методов к преподаванию учебной дисциплины**

При организации образовательного процесса используются

##### **1) эвристический подход:**

- осуществление студентами лично-значимых открытий окружающего мира;
- демонстрация многообразия решений большинства профессиональных задач и жизненных проблем;

- творческую самореализацию обучающихся в процессе создания образовательных продуктов;

- индивидуализация обучения через возможность самостоятельно ставить цели, осуществлять рефлексию собственной образовательной деятельности;

**2) *практико-ориентированный подход:***

- освоение содержания образования через решения практических задач;  
- приобретение навыков эффективного выполнения разных видов профессиональной деятельности;

- ориентация на генерирование идей, реализацию групповых студенческих проектов, развитие предпринимательской культуры;

- использование процедур, способов оценивания, фиксирующих сформированность профессиональных компетенций.

**3) *метод проектного обучения:***

- способ организации учебной деятельности студентов, развивающий актуальные для учебной и профессиональной деятельности навыки планирования, самоорганизации, сотрудничества и предполагающий создание собственного продукта;

- приобретение навыков для решения исследовательских, творческих, социальных, предпринимательских и коммуникационных задач.

**4) *методы и приемы развития критического мышления,*** которые представляют собой систему, формирующую навыки работы с информацией в процессе чтения и письма; понимания информации как отправного, а не конечного пункта критического мышления.

**5) *метод группового обучения,*** который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

### **Методические рекомендации по организации самостоятельной работы обучающихся**

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

– изучение литературы и материалов электронных источников по проблемам дисциплины;

– выполнение домашнего задания;

– подготовка к лабораторным занятиям;

– курсовые, дипломные и научно-исследовательские работы, связанные с тематикой дисциплины;

– подготовка к участию в конференциях с докладами по проблемам дисциплины.

Для организации дистанционной и самостоятельной работы студентов по учебной дисциплине рекомендуется использовать современные информационные ресурсы, размещенные на образовательном портале смешанного и дистанционного обучения БГУ:

**<https://edummf.bsu.by/course/view.php?id=619>**,

<https://edummf.bsu.by/course/view.php?id=1100>,

<https://edummf.bsu.by/course/view.php?id=1064>.

и содержащие учебные материалы (курс лекций, задания к лабораторным работам, вопросы и задачи к экзамену, тесты и т.п.) для электронного сопровождения изучаемой дисциплины.

### Примерный перечень вопросов к экзамену

#### Основы веб-технологий.

1. Протоколы интернет tcp/ip, http, https, ftp. Характеристика, назначение.
2. Что такое HTTP? Как работает протокол HTTP? Укажите методы HTTP-запросов.
3. Что такое сессии и куки? В чем основные различия между ними?
4. Какие задачи выполняет каждая из технологии html, css, js, php?
5. Различие между статическим и динамическим контентом на веб-странице.
6. Подходы и методики разработки верстки сайта (mobileFirst/desktopFirst graceful degradation/progressive enhancement)

#### JavaScript:

7. Какова роль JavaScript в веб-разработке?
8. Разница между let, const и var в JavaScript при объявлении идентификаторов.
9. Типы данных в JavaScript. Передача параметров.
10. Иммутабельность примитивных данных.
11. Что такое DOM и как он используется в JavaScript? События в JavaScript
12. Навигация по DOM-элементам. Методы изменения DOM.
13. Способы назначения обработчиков событий. Фазы всплытие/погружение событий. Делегирование событий.
14. Назначение знака плюс в Js. Преобразование типов.
15. Методы массивов в Js. Какие методы мутируют массив?
16. Перебираемые (итерируемые) объекты. Отличие от массивов.
17. Объекты в JavaScript. Методы объектов.
18. Классы в Js. Конструкторы. Методы.
19. Асинхронность в Js, концепция event-loop. Сетевые запросы. Fetch.

#### Фронтенд-фреймворки и библиотеки:

20. Какие фронтенд-фреймворки и библиотеки вы знаете? В чем разница между ними?
21. Опишите основные особенности Angular, React и Vue.js.
22. Библиотека React. Идеология React. Основные концепции React
23. JSX в React. Компоненты в React. Виртуальный DOM и Fiber.
24. Рендеринг компонент в React. Какие хуки используются для эмуляции методов жизненного цикла в функциональных компонентах?
25. Хуки. Базовые хуки в React. Использование useState и useEffect.
26. Состояние в компоненте React. Управление состоянием. Иммутабельность.



27.Опишите несколько подходов к использованию и хранению данных в компонентах.

**Бэкенд-разработка:**

28.Что такое сервер и клиент в веб-разработке?

29.Какие языки программирования широко используются для бэкенд-разработки?

30.Серверная технология PHP. Скрипты php. Переменные, типы данных.Основные отличия синтаксиса php от js.

31.Как работает база данных в веб-разработке? Виды запросов в СУБД.

**Работа с базами данных:**

32.Что такое SQL и NoSQL базы данных? Приведите примеры.

33.Как подключить БД к серверной части веб-приложения.

34.Какие операции можно выполнять с базой данных с использованием SQL?

## ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Методы программирования	ВТиКМ	нет	Вносить изменения не требуется (протокол № 11 от 24.05.2023)
Технологии программирования	ВТиКМ	нет	Вносить изменения не требуется (протокол № 11 от 24.05.2023)

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ**  
на \_\_\_\_\_ / \_\_\_\_\_ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры Веб-технологий и компьютерного моделирования (протокол № \_\_\_ от \_\_\_ 202\_ г.)

Заведующий кафедрой

кандидат физ.-мат. наук, доцент \_\_\_\_\_

М.В. Игнатенко

УТВЕРЖДАЮ

Декан факультета

доктор физ.-мат. наук, доцент \_\_\_\_\_

С.М. Босяков