**Приложение 4**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное**
**образовательное учреждение высшего образования**
**«Национальный исследовательский Нижегородский государственный университет**
**им. Н.И. Лобачевского»**

Павловский филиал ННГУ

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

**Рабочая программа дисциплины**

Интернет-программирование

Уровень высшего образования

Бакалавриат

Направление подготовки / специальность

09.03.03 - Прикладная информатика

Направленность образовательной программы

Прикладная информатика в экономике и управлении

Форма обучения

очная, очно-заочная

г. Павлово

2024 год начала подготовки

## 1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.12 Интернет-программирование относится к части, формируемой участниками образовательных отношений образовательной программы.

## 2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

| **Формируемые компетенции** (код, содержание компетенции) | **Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции** | | **Наименование оценочного средства** | |
|---|---|---|---|---|
| | **Индикатор достижения компетенции** (код, содержание индикатора) | **Результаты обучения по дисциплине** | **Для текущего контроля успеваемости** | **Для промежуточной аттестации** |
| *ПК-11: Способен осуществлять модульное и интеграционное тестирование ИС (ИИС), устранять (по мере возможности) обнаруженные несоответствия* | *ПК-11.1: Демонстрирует знание методологических основ модульного и интеграционного тестирования ИС (ИИС)*<br>*ПК-11.2: Демонстрирует умение осуществлять модульное и интеграционное тестирование ИС (ИИС) и устранять (по мере возможности) обнаруженные несоответствия*<br>*ПК-11.3: Имеет практический опыт модульного и интеграционного тестирования конкретной ИС (ИИС)* | *ПК-11.1:*<br>*Знать структуру языка гипертекстовой разметки HTML, технологию создания гипертекстовых докумен-тов*<br><br>*ПК-11.2:*<br>*Уметь выполнять создание и оформление связанного набора web-страниц, навыками построения шаблона страницы*<br><br>*ПК-11.3:*<br>*Владеть: навыками тестирования ИС* | *Контрольная работа*<br>*Тест* | *Экзамен:*<br>*Контрольные вопросы* |
| *ПК-9: Способен моделировать прикладные (бизнес) процессы и объекты предметной области* | *ПК-9.1: Демонстрирует знание методических основ моделирования процессов и объектов предметной области*<br>*ПК-9.2: Демонстрирует умение применения знаний к моделированию прикладных процессов и объектов предметной области при разработке программного обеспечения ИС*<br>*ПК-9.3: Имеет практический опыт моделирования процессов и объектов на примере конкретной предметной* | *ПК-9.1:*<br>*Знать структуру языка гипертекстовой разметки HTML, технологию создания гипертекстовых документов*<br><br>*ПК-9.2:*<br>*Уметь выполнять создание и оформление связанного набора web-страниц*<br><br>*ПК-9.3:*<br>*Владеть: навыками построения шаблона страницы* | *Контрольная работа*<br>*Тест* | *Экзамен:*<br>*Контрольные вопросы* |

| | | | | |
|---|---|---|---|---|
| | *области* | | | |

## 3. Структура и содержание дисциплины

### 3.1 Трудоемкость дисциплины

| | **очная** | **очно-заочная** |
|---|---|---|
| **Общая трудоемкость, з.е.** | **4** | **4** |
| **Часов по учебному плану** | **144** | **144** |
| в том числе | | |
| **аудиторные занятия (контактная работа):** | | |
| **- занятия лекционного типа** | **14** | **12** |
| **- занятия семинарского типа (практические занятия / лабораторные работы)** | **28** | **12** |
| **- КСР** | **2** | **2** |
| **самостоятельная работа** | **64** | **82** |
| **Промежуточная аттестация** | **36**<br>**Экзамен** | **36**<br>**Экзамен** |

### 3.2. Содержание дисциплины

*(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)*

| Наименование разделов и тем дисциплины | Всего (часы) | | в том числе | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Контактная работа (работа во взаимодействии с преподавателем), часы из них | | | | | | Самостоятельная работа обучающегося, часы | |
| | | | Занятия лекционного типа | | Занятия семинарского типа (практические занятия/лабораторные работы), часы | | Всего | | | |
| | очно | очно-заочно | очно | очно-заочно | очно | очно-заочно | очно | очно-заочно | очно | очно-заочно |
| Тема 1. Ведение в курс. История сети Internet. Тенден-ции развития программного обеспечения для Интернета. | 12 | 12 | 2 | 1 | 1 | 1 | 3 | 2 | 9 | 10 |
| Тема 2. Язык разметки HTML. Модель OSI. Протокол Ethernet. DNS. | 15 | 15 | 2 | 1 | 4 | 1 | 6 | 2 | 9 | 13 |
| Тема 3. Протокол TCP, Протокол HTTP. E-mail. Язык стилей CSS. | 17 | 17 | 2 | 2 | 6 | 2 | 8 | 4 | 9 | 13 |
| Тема 4 Объектная модель документа Web-серверы. Серверная разработка. Front-end и Back-end | 16 | 16 | 2 | 2 | 5 | 2 | 7 | 4 | 9 | 12 |
| Тема 5. Язык JavaScript | 16 | 16 | 2 | 2 | 5 | 2 | 7 | 4 | 9 | 12 |
| Тема 6. Библиотека JQuery | 16 | 16 | 2 | 2 | 5 | 2 | 7 | 4 | 9 | 12 |
| Тема 7. Технологии AJAX. | 14 | 14 | 2 | 2 | 2 | 2 | 4 | 4 | 10 | 10 |
| Аттестация | 36 | 36 | | | | | | | | |
| КСР | 2 | 2 | | | | | 2 | 2 | | |
| Итого | 144 | 144 | 14 | 12 | 28 | 12 | 44 | 26 | 64 | 82 |

**Содержание разделов и тем дисциплины**

1. Введение в информатику.
Основы архитектуры и организации ЭВМ
Базовые представления об архитектуре ЭВМ. История развития вычислительных средств. Принципы фон Неймана. Классификация компьютеров. Архитектура современного компьютера
Архитектура микропроцессора
Классы процессоров. Конвейерная обработка команд. Векторная обработка. Регистры процессоров. Системы команд x86. Макроассемблер.
Запоминающие устройства. Иерархия запоминающих устройств. Устройства ввода и вывода
Организация оперативной памяти компьютера.
Логическая память. Сегменты. Связывание адресов. Простейшие схемы управления памятью. Схемы с фиксированными и переменными разделами. Страничная память. Сегментно-страничная организация памяти.
Арифметические основы ЭВМ
Системы счисления. Представление чисел в позиционных системах счисления. Обоснование экономичности систем счисления.
Представление целых и вещественных чисел в позиционных системах счисления.
Арифметические операции над числами, представленными в двоичной системе счисления.
Способы представления чисел в ЭВМ: с фиксированной запятой, с плавающей запятой.
Машинные методы выполнения арифметических операций над числами, представленными в двоичной системе счисления: преобразование кодов (прямой, обратный, дополнительный).
Умножение с младших разрядов и старших разрядов в прямом коде. Деление с восстановлением остатка и без восстановления остатка.
Проблема переполнения. Ошибка усечения. Десятичные двоично-кодированные системы.
Логические основы ЭВМ
Алгебра высказываний. Операции над высказываниями. Логические связки: инвер-сия, дизъюнкция, конъюнкция, импликация, эквиваленция, сложение по mod 2. Ранги логических связок.
Таблицы истинности. Правила составления таблиц истинности.
Формулы алгебры высказываний и равносильные преобразования. Основы равносильности.
Равносильности, выражающие одни логические операции через другие.
Функции алгебры логики (ФАЛ). ФАЛ одной и двух переменных.
Дизъюнктивные и конъюнктивные нормальные формы алгебры высказываний. СДНФ. СКНФ.
Минимизация ФАЛ: метод непосредственных преобразований, метод графической минимизации Карно.
Типовые логические элементы и узлы ЭВМ: «И», «ИЛИ», «НЕ», «И-НЕ», «ИЛИ-НЕ», «триггер», «полусумматор».
Системное программное обеспечение
1. Базовое программное обеспечение. Служебные программы.
2. Операционные системы.
2.1. Назначение операционной системы. Виды операционных систем. Базовые понятия операционных систем. Процессы и потоки. Управление памятью. Драйверы устройств. Файловые системы.
2.2. Архитектура Windows.
2.3. Архитектура Linux.
Прикладное программное обеспечение.
Текстовый процессор Word
Ввод и редактирование текста. Ввод текста с использованием “горячих” клавиш. Списки. Таблицы. Создание рисунков с помощью инструментов Word. Формулы и вставка символов. Вставка рисунков в текст. Шаблоны и формы. Построение диаграмм. Документы слияния. Главный документ. Исправления. Автозамена. Примечания. Сноски.
Табличный процессор Excel
Создание и редактирование таблиц. Диаграммы. Формы и сортировка таблиц. Консолидация данных.

Сводная таблица. Использования финансовых функций в формулах таблицы. Подбор параметра. Поиск решения. Шаблоны и элементы управления на рабочих листах. Функции пользователя. Элементы управления на листах – формы.
Основы информационных систем. Базы данных
Основные понятия Классификация БД Модели данных. Проектирование баз данных. CASE- системы для разработки информационных сметем. Реляционная модель данных. Получение реляционной схемы из инфологической модели.
База данных Microsoft Access. Потенциальные и внешние ключи. Связывание таблиц. Нормализация таблиц. Ограничения целостности.
Компьютерные сети.
1. Сетевые архитектуры
1.1. Назначение и классификация компьютерных сетей. Типы сетей. Топология сетей. Сетевые компоненты. Сети Ethernet. Сети Token Ring. Сетевые протоколы. Среда клиент-сервер. Internet как иерархия сетей.
2. Сетевые информационные технологии
2.1. История создания Internet. Административное устройство. Основные понятия (гипертекст, Web, http, URL, IP – адрес, доменное имя, браузер).
2.2. Язык разметки гипертекста HTML.
2.2.1. История развития. Принцип гипертекстовой разметки. Структура документа HTML. Элементы структурной организации текста. Элементы логического форматирования символов. Элементы физического форматирования символов. Ссылки. Списки. Таблицы. Вставка в документ объектов. Элементы , , Элемент . Фреймы.
2.3. Введение в CSS. Слои. Введение в HTML5.
Основные понятия языков программирования. Сравнительная характеристика языков программирования. Синтаксис, семантика языков программирования. Сравнение развития языков в представлении данных и способах реализации алгоритмов. Общая характеристика языка программирования. Основные элементы языка программирования: алфавит, идентификаторы, ключевые слова, константы, переменные, операторы, выражения, функции. Общая структура программы. Исполняемые инструкции и определения. Пример простой программы.
Особенности программирования в современных оконных операционных средах. Среда разработки. Порядок создания консольного приложения.
Элементы модульного программирования. Многофайловая организация про-грамм. Состав файлов проекта. Подключение библиотек.
Процесс обработки программы на языке высокого уровня: компиляция, сборка, выполнение и отладка.
2. Понятие типа данных. Целочисленные и вещественные типы данных Представление в ЭВМ знаковых и беззнаковых целочисленных типов Характеристики целочисленных типов данных: ключевые слова, размер, диапазон представления. Операции над данными целого типа. Представление в ЭВМ вещественных типов данных. Характеристики вещественных типов данных. Операции над данными вещественных типов. Преобразование типа. Вычисление значений выражений.
3. Алгоритмизация процессов обработки данных.
Введение в алгоритмы обработки данных. Понятие алгоритма и его свойства. Основные конструкции для записи алгоритмов. Базисные алгоритмы: следование, выбор, повторение. Структурное программирование
Представление базисных алгоритмов на языке высокого уровня Управляющие инструкции. Условный оператор. Логические выражения. Переключатель. Операторы цикла с предусловием и постусловием. Вложенные циклы
Простейшие алгоритмы обработки данных. Вычисление значений выражений. Использование логических выражений в задачах на выделение геометрических областей. Итерационные алгоритмы. Циклы с известным и неизвестным числом повторений. Сочетание цикла и разветвлений. Рекуррентные вычисления. Алгоритмы поиска корней функции. Применение циклов в задачах на темы: суммирование

рядов, позиционная запись числа, делители целого числа.
Одномерные массивы данных. Построение и преобразование одномерных массивов. Алгоритмы обработки массивов данных. Последовательный поиск. Поиск максимального и минимального элементов. Построение массива без повторений. Бинарный поиск в упорядоченном массиве. Алгоритмы сортировки. Операции с многочленами, заданными массивами своих коэффициентов.
Построение и преобразование матриц. Матричная алгебра.
Форматный ввод вывод. Функции форматного ввода вывода стандартной библиотеки.
Символьный тип данных. Код символа. Кодовая таблица. Символьная константа. Ввод и вывод символов
4. Функции.
Описание функции. Возврат значений. Передача параметров. Массивы как аргументы функций.
Строковый тип данных. Строковые константы. Ввод и вывод строк. Функции для работы со строками.
Классы памяти. Внешние, статические, автоматические, регистровые переменные. Правила областей действия.
Структуры данных стек, очередь. Непрерывная реализация структур данных на базе линейной памяти ЭВМ. Пример использования стека в программе-калькулятор.
Препроцессор. Включение файлов. Макроподстановка. Условная компиляция.
Модульное программирование. Многофайловая организация программы-калькулятор.
Рекурсивные алгоритмы обработки данных. Условия, обеспечивающие завершение рекурсивных вызовов. Реализация рекурсивных вызовов. Сравнение рекурсивных и итерационных алгоритмов обработки данных.
Указатели. Указатели и адреса. Указатели и аргументы функций. Указатели и массивы Адресная арифметика. Указатели и строки. Динамическое выделение памяти.
Массивы указателей. Сортировка текстовых строк входного потока. Многомерные массивы. Массивы и функции. Двумерные массивы и указатели. Функция, суммирующая матрицы произвольной размерности.
Указатели на функции. Массивы указателей на функции. Сложные типы данных. Разработка полиморфных программ.
5. Структуры.
Расположение в памяти. Инициализация. Операции. Приоритет операций. Вложенные структуры.
Объединения и поля битов. Структуры и функции. Массивы структур. Указатели на структуры
6. Файловый ввод - вывод
Современные средства языка программирования. Оператор разрешения области видимости.
Использование функции в качестве элементов структур. Объявления в операторах. Константные объекты. Встроенные функции. Параметры по умолчанию. Передача аргументов функциям и возврат значений по адресу. Перегрузка имен функций
7. Введение в объектно-ориентированное программирование.
Стили программирования. Принципы объектно-ориентированного программирования. Абстракция данных. Инкапсуляция. Наследование. Полиморфизм. Повторное использование кода.
Классы.
Инкапсуляция и члены-функции. Управление доступом. Конструкторы и деструкторы. Отличительные свойства. Взаимодействие с компилятором.
Копирование объектов классов. Массивы объектов классов. Объекты классов как элементы
Примеры разработки классов: строка, вектор, матрица.

8. Объектно-событийное программирование в операционной среде. Событие и сообщение. Кодирование сообщений и механизмы реализации обмена сообщениями в операционной среде. Программирование, управляемое событиями. Природа событий. Виды событий. События от мыши. События от клавиатуры. Передача сообщений.
Визуальное программирование в оконной операционной среде. Обзор современных инструментальных

систем визуального программирования. Основные характеристики среды. Настройка среды. Основные элементы интегрированной среды разработки. Основы визуального программирования Методы визуальной разработки оконных приложений. Выбор шаблона приложения. Работа с формой. Стандартные визуальные компоненты. Перенос компонент на форму. Установка свойств. События, на которые реагируют компоненты. Задание обработчиков событий Примеры разработки оконных приложений. Организация ввода вывода данных. Разработка программы по обработке простых типов данных. Приложение с кнопками, редактируемым полем и списком. (С помощью кнопок можно добавлять элементы в список, удалять из списка) Приложение - простейший калькулятор

Практические занятия /лабораторные работы организуются, в том числе, в форме практической подготовки, которая предусматривает участие обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.
На проведение практических занятий / лабораторных работ в форме практической подготовки отводится: очная форма обучения - 21 ч., очно-заочная форма обучения - 12 ч.

**4. Учебно-методическое обеспечение самостоятельной работы обучающихся**

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Цель самостоятельной работы - формирование навыков непрерывного самообразования и профессионального совершенствования.
Самостоятельная работа способствует формированию аналитического и творческого мышления, совершенствует способы организации исследовательской деятельности, воспитывает целеустремленность, системность и последовательность в работе студентов, развивает у них навык завершать начатую работу.
Основные виды самостоятельной работы студентов:
- работа с основной и дополнительной литературой;
- изучение категориального аппарата дисциплины;
- самостоятельное изучение тем дисциплины;
- подготовка к экзамену;
- работа в библиотеке;
- изучение сайтов по темам дисциплины в сети Интернет.

Работа с основной и дополнительной литературой
Изучение рекомендованной литературы следует начинать с учебников и учебных пособий, затем переходить к научным монографиям и материалам периодических изданий. Работа с литературой предусматривает конспектирование наиболее актуальных и познавательных материалов. Это не только мобилизует внимание, но и способствует более глубокому осмыслению материала, его лучшему запоминанию, а также позволяет студентам проводить систематизацию и сравнительный анализ изучаемой информации. Таким образом, конспектирование – одна из основных форм самостоятельного труда, которая требует от студента активно работать с учебной литературой и не ограничиваться конспектом лекций. Студент должен уметь самостоятельно подбирать необходимую литературу для учебной и научной работы, уметь обращаться с предметными каталогами и библиографическим справочником библиотеки.
Изучение категориального аппарата дисциплины
Изучение и осмысление экономических категорий требует проработки лекционного материала,

выполнения практических заданий, изучение словарей, энциклопедий, справочников.
Индивидуальная самостоятельная работа студента направлена на овладение и грамотное применение экономической терминологии в области компьютерного моделирования.
Самостоятельное изучение тем дисциплины
Особое место отводится самостоятельной проработке студентами отдельных разделов и тем изучаемой дисциплины. Такой подход вырабатывает у студентов инициативу, стремление к увеличению объема знаний, умений и навыков, всестороннего овладения способами и приемами профессиональной деятельности.
Изучение вопросов определенной темы направлено на более глубокое усвоение основных категорий экономической теории, понимание экономических процессов, происходящих в обществе, совершенствование навыка анализа теоретического и эмпирического материала.
Подготовка к экзамену
Промежуточная аттестация студентов по дисциплине проходит в виде экзамена и предусматривает оценку. Условием успешного прохождения промежуточной аттестации является систематическая работа студента в течение семестра. В этом случае подготовка к экзамену является систематизацией всех полученных знаний по данной дисциплине.
Рекомендуется внимательно изучить перечень вопросов к экзамену, а также использовать в процессе обучения программу, учебно-методический комплекс, другие методические материалы.
Желательно спланировать троекратный просмотр материала перед экзаменом. Во-первых, внимательное чтение с осмыслением, подчеркиванием и составлением краткого плана ответа. Во-вторых, повторная проработка наиболее сложных вопросов. В-третьих, быстрый просмотр материала или планов ответов для его систематизации в памяти.
Самостоятельная работа в библиотеке
Важным аспектом самостоятельной подготовки студентов является работа с библиотечным фондом.
Это работа предполагает различные варианты повышения профессионального уровня студентов:
а) получение книг для подробного изучения в течение семестра на научном абонементе;
б) изучение книг, журналов, газет - в читальном зале;
в) возможность поиска необходимого материала посредством электронного каталога;
г) получение необходимых сведений об источниках информации у сотрудников библиотеки.
Изучение сайтов по темам дисциплины в сети Интернет
Ресурсы Интернет являются одним из альтернативных источников быстрого поиска требуемой информации. Их использование возможно для получения основных и дополнительных сведений по изучаемым материалам. Необходимо помнить об оформлении ссылок на Интернет-источники.
Для повышения эффективности самостоятельной работы студентов преподавателю целесообразно использовать следующие виды деятельности:
- консультации,
- выдача заданий на самостоятельную работу,
- информационное обеспечение обучения,
- контроль качества самостоятельной работы студентов.

**5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)**

**5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:**

**5.1.1 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-11:**

**Вариант 1**

**1**. Создать HTML-страницу, которая при загрузке случайным образом выводит один из трех эпиграфов.

**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, соблюдается баланс открытых и закрытых скобок. При этом необходимо учесть, что закрывающая скобка не может появиться раньше соответствующей ей открывающей скобки. Функция возвращает строку "O'key" при соблюдении такого баланса, и "Error" в противном случае.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Добавьте к системному объекту **Date** метод **isBusinessTime**. Ваш метод, должен возвращать true, если экземпляр Date, в контексте которой он вызван, задает рабочее время и false, в противном случае.

Рабочим временем считайте время с 8 до 17 часов во все дни, кроме субботы и воскресенья.

Напишите программу с тестами для демонстрации работы Вашего метода.

**Вариант 2.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, выпадает ли 13-ое число текущего месяца на пятницу.

**2**. Написать функцию, которая в строке, переданной в качестве аргумента, выделяет наибольшую подстроку (если она существует), заключенную между символами '**(**' и '**)**'.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Для объекта **String** добавить метод **stripLast**(), который удаляет символы пробела в конце строки и возвращает полученную строку. Так, например, для строки

"  Задание 306  "

данный метод должен вернуть
"  Задание 306"

**Вариант 3.**

**1**. Создать HTML-страницу, которая выводит коэффициент Ваших интеллектуальных способностей *I(l)* на текущий день, используя формулу

I=cos(l)/l^2,

где *l* - количество дней, прошедших от Вашего рождения.

**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, передается целое число в диапазоне от 0 до 1000. Если это так, то функция должна вернуть *True*, в противном случае - *False*.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста и кнопкой **submit**. В эти поля будут вводиться адреса электронной почты.

У формы должно обрабатываться событие **submit**. Обработка состоит в следующем:

1. если форма заполнена правильно *(все поля заполнены и во всех стоят адреса вида ??? @???????, где знак "?" может быть любой латинской буквой, точкой, минусом или подчерком)*;
   a. удалить все пробелы;
   b. выдать сообщение, что все в порядке (**alert**);
2. если форма заполнена неправильно
   a. выдать сообщение об ошибке;
   b. установить курсор в первое ошибочное поле.

Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 4.**

**1**. Напечатайте количество часов в текущем веке.

**2**. Написать функцию, которая возвращает количество слов в строке, переданной в качестве аргумента.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста. Далее обеспечьте такую возможность: если человек набрал в некотором поле число и перешел в другое поле, в оставляемом поле число должно автоматически преобразоваться в "денежный" вид - pppp.кк
Например, если набрано 12, должно получиться 12.00, набрано 1.3, должно получиться 1.30, и т.д.
Если же то, что было набрано, не является правильным числом, Ваша программа должна снова поставить курсор в ошибочное поле для внесения исправлений.
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 5.**

**1**. Создать HTML-страницу, на которой была бы случайная картинка из некоторого набора. Возьмите любые картинки.
**2.** Для объекта **Date** добавить метод **leap**(), который возвращает true, если год високосный и false - если нет.
**3.** Сделайте страницу, на которой присутствует список (**SELECT SINGLE**) с названиями (*не URL!!!*) нескольких страниц и кнопка.
Человек может набрать название страницы из списка и нажать кнопку. Должна загрузиться выбранная страница. Если в списке ничего не было выбрано, выдайте сообщение об ошибке (**alert**).

**Вариант 6.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней осталось до стипендии, если ее выдают 25 числа каждого месяца за исключением месяцев летних каникул. При этом в день выдачи стипендии нужно сообщать **Сегодня дадут стипендию!**, а на следующий день после стипендии - **Еще целый месяц до стипендии!**.
**2**. Для объекта **Date** добавить метод **firstDayOfMonth()**, который возвращает строку с днем недели (по-русски) первого дня месяца данного объекта.
**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста и кнопкой **submit**. В эти поля будут вводиться названия городов.
У формы должно обрабатываться событие **submit**. Обработка состоит в следующем:

1. если форма заполнена правильно *(все поля заполнены и ни одно не пусто)*;
a. удалить все пробелы;
b. выдать сообщение, что все в порядке (**alert**);

2. если форма заполнена неправильно
a. выдать сообщение об ошибке;
b. установить курсор в первое ошибочное поле.

Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 7.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней прошло с начала учебного года.
**2**. Добавьте к системному объекту **String** метод **replaceLast** с двумя параметрами. Первый параметр - "что искать", второй - "на что заменить". Ваш метод должен возвращать строку, в контексте которой вызван и в которой последнее вхождение подстроки "что искать" заменено на подстроку "на что заменить".

Напишите программу с тестами для демонстрации работы Вашего метода.
**3**. Возьмите несколько (не менее пяти) любых картинок. Сделайте так, чтобы на Вашей странице отображались одновременно все картинки в различных местах и около каждой картинки был изначально "включенный" чек-бокс. С помощью чек-боксов пользователь должен иметь возможность включать/выключать показ соответствующих картинок. Т.е. немедленно по нажатии чек-бокса картинка должна исчезать (или появляться).
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 8.**

**1**. Создать HTML-страницу, которая при загрузке выводит названия трех случайных карт по одному на строке. Сначала достоинство, а потом масть.
Например:
Тройка бубей
Семерка треф
Дама пик
**2**. Для объекта **String** добавить метод **isDigits()**, который возвращает true, если строка состоит только из цифр, и false - в противном случае.
**3**. Сделайте "осторожный" **reset**. На вашей странице должна быть форма с различными элементами *(не менее пяти)*, такими как поле ввода, чек-бокс, список и т.д. При нажатии кнопки **reset**. Ваша программа - обработчик этого события должна проверять, изменено ли что-либо в форме или все осталось, как было изначально. Если форма изменена, пользователю должен быть задан вопрос (функция **confirm**): "Вы собираетесь сбросить форму. При этом, все, что Вы ввели будет потеряно. Сбросить форму?". Далее форма должна быть сброшена или не сброшена в зависимости от ответа пользователя.

**Вариант 9.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней прошло с 9 мая 1945 года.
**2**. В объект **Array** добавьте метод **notZero()**, который возвращает количество элементов массива, не являющихся нулями.
**3**. Возьмите пять произвольных картинок. Разместите на экране группу кнопок radiobutton. Каждая из кнопок соответствует одной картинке. В зависимости от того, какую опцию выбрал пользователь на экране появляется та или иная картинка.

**Вариант 10.**

**1**. Создать HTML-страницу со случайным цветам текста.
**2**. Написать функцию, которая в строке, переданной в качестве аргумента, заменяет вхождение двух подряд идущих пробелов на один пробел и возвращает полученную строку.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Сделайте на своей странице будильник. Человек, вошедший на нее, должен иметь возможность указать в некотором текстовом поле желаемое время. В указанное время должно

появиться сообщение (**alert**). Если Вам так удобнее, сделайте, чтобы после ввода времени он должен был нажать кнопку "Ввод".

**Вариант 11.**

**1**. Создать HTML-страницу, которая при загрузке выводит таблицу размерами 3x3, каждая клетка которой заполняется случайным образом либо символом "+", либо символом "o".

**2**. Для объекта **Date** добавить метод **MonthName()**, который возвращает строку с названием месяца по-русски ("Январь", "Февраль" и т.д.).

Добавьте к системному объекту **Date** метод **print**. Ваш метод, должен возвращать строку для печати. Строка, будучи напечатана с помощью document.write, должна содержать дату по-русски черным цветом, если **Date** задает рабочий день, и красным - если субботу или воскресенье.

Напишите программу с тестами для демонстрации работы Вашего метода.

**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста. У формы должно обрабатываться событие **submit**. Обработка состоит в следующем: если все поля формы заполнены (не пусты), выдается сообщение (**alert**) "О'кей". Если же хотя бы одно поле пустое, выдается сообщение об ошибке и, как только человек нажмет "Ok", курсор устанавливается в ошибочное поле.

Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 12.**

**1**. Создать HTML-страницу, которая печатает Ваш возраст в годах, месяцах и днях.

**2**. Написать функцию, которая возвращает количество символов 'a' в строке, переданной в качестве аргумента.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поля ввода с именем "In" и с именем "Result";
2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:

ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"

где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Сделайте "навязчивую" ссылку. Это обычная ссылка на другую страницу, но, если пользователь нацелит на нее мышку, а затем, не нажав на ссылку, попытается убрать мышку в сторону, будет выдано сообщение: "Эй!!! А нажать на меня? Забыл? Нажать сейчас?". Это сообщение должно выдаваться функцией **confirm**, чтобы человек мог ответить "Да" или "Нет". Если ответ - "Да", должен быть осуществлен переход так, как если бы он нажал на ссылку. Разместите несколько таких ссылок на странице.

**Вариант 13.**

**1**. Создать HTML-страницу, которая при загрузке выводит названия комбинацию игры в кости - два целых случайных числа от 1 до 6.

**2**. Написать функцию, которая проверяет, состоит ли строка из символов “*”.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поля ввода с именем "In" и с именем "Result";
2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:

ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Добавьте к системному объекту **String** метод **find** с одним параметром – строкой “что найти”. Ваш метод должен возвращать число вхождений параметра в строку, в контексте которой вызван. Напишите программу с тестами для демонстрации работы Вашего метода.

**Вариант 14.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней осталось до первого января 2014 года.
**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, введена хотя бы одна цифра. Если это так, то функция возвращает *true*, в противном случае - *false*. Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1 поля ввода с именем "In" и с именем "Result";
2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. В объект **Array** добавьте метод **notEmpty()**, который возвращает количество элементов массива, не являющихся пустыми строками.

**Вариант 15.**

**1**. Создать HTML-страницу, которая выводит коэффициент Ваших физических способностей *F(l)* на ближайший выходной, используя формулу

I=l^3/sin(l),
где *l* - количество дней, прошедших от Вашего рождения.
**2**. Написать функцию, которая определяет пол человека по его имени, отчеству и фамилии. Для этого Вам необходимо создать форму с именем "Test", в которой помещены:
1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Сделайте страницу, на которой сначала не видно ни одной картинки, она выглядит пустой. Однако стоит набрать на клавиатуре слово "show", как картинки немедленно появляются. Если потом набрать "hide", они исчезают.
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**5.1.2 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-9:**

**Вариант 1**

**1**. Создать HTML-страницу, которая при загрузке случайным образом выводит один из трех эпиграфов.

**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, соблюдается баланс открытых и закрытых скобок. При этом необходимо учесть, что закрывающая скобка не может появиться раньше соответствующей ей открывающей скобки. Функция возвращает строку "O'key" при соблюдении такого баланса, и "Error" в противном случае.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Добавьте к системному объекту **Date** метод **isBusinessTime**. Ваш метод, должен возвращать true, если экземпляр Date, в контексте которой он вызван, задает рабочее время и false, в противном случае.
Рабочим временем считайте время с 8 до 17 часов во все дни, кроме субботы и воскресенья.

Напишите программу с тестами для демонстрации работы Вашего метода.

**Вариант 2.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, выпадает ли 13-ое число текущего месяца на пятницу.
**2**. Написать функцию, которая в строке, переданной в качестве аргумента, выделяет наибольшую подстроку (если она существует), заключенную между символами '**(**' и '**)**'.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Для объекта **String** добавить метод **stripLast**(), который удаляет символы пробела в конце строки и возвращает полученную строку. Так, например, для строки
"  Задание 306  "
данный метод должен вернуть
"  Задание 306"

**Вариант 3.**

**1**. Создать HTML-страницу, которая выводит коэффициент Ваших интеллектуальных способностей *I(l)* на текущий день, используя формулу

I=cos(l)/l^2,
где *l* - количество дней, прошедших от Вашего рождения.
**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, передается целое число в диапазоне от 0 до 1000. Если это так, то функция должна вернуть *True*, в противном случае - *False*.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
   ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
   где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста и кнопкой **submit**. В эти поля будут вводиться адреса электронной почты.
У формы должно обрабатываться событие **submit**. Обработка состоит в следующем:

1. если форма заполнена правильно *(все поля заполнены и во всех стоят адреса вида ??? @???????, где знак "?" может быть любой латинской буквой, точкой, минусом или подчерком)*;
   a. удалить все пробелы;
   b. выдать сообщение, что все в порядке (**alert**);
2. если форма заполнена неправильно
   a. выдать сообщение об ошибке;
   b. установить курсор в первое ошибочное поле.

Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 4.**

**1**. Напечатайте количество часов в текущем веке.
**2**. Написать функцию, которая возвращает количество слов в строке, переданной в качестве аргумента.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста. Далее обеспечьте такую возможность: если человек набрал в некотором поле число и перешел в другое поле, в оставляемом поле число должно автоматически преобразоваться в "денежный" вид - pppp.кк
Например, если набрано 12, должно получиться 12.00, набрано 1.3, должно получиться 1.30, и т.д.
Если же то, что было набрано, не является правильным числом, Ваша программа должна снова поставить курсор в ошибочное поле для внесения исправлений.
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 5.**

**1**. Создать HTML-страницу, на которой была бы случайная картинка из некоторого набора. Возьмите любые картинки.
**2.** Для объекта **Date** добавить метод **leap**(), который возвращает true, если год високосный и false - если нет.
**3.** Сделайте страницу, на которой присутствует список (**SELECT SINGLE**) с названиями (*не URL!!!*) нескольких страниц и кнопка.
Человек может набрать название страницы из списка и нажать кнопку. Должна загрузиться выбранная страница. Если в списке ничего не было выбрано, выдайте сообщение об ошибке (**alert**).

**Вариант 6.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней осталось до стипендии, если ее выдают 25 числа каждого месяца за исключением месяцев летних каникул. При этом в день выдачи стипендии нужно сообщать **Сегодня дадут стипендию!**, а на следующий день после стипендии - **Еще целый месяц до стипендии!**.
**2**. Для объекта **Date** добавить метод **firstDayOfMonth()**, который возвращает строку с днем недели (по-русски) первого дня месяца данного объекта.
**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста и кнопкой **submit**. В эти поля будут вводиться названия городов.
У формы должно обрабатываться событие **submit**. Обработка состоит в следующем:

1. если форма заполнена правильно *(все поля заполнены и ни одно не пусто)*;
a. удалить все пробелы;
b. выдать сообщение, что все в порядке (**alert**);

2. если форма заполнена неправильно
a. выдать сообщение об ошибке;
b. установить курсор в первое ошибочное поле.

Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 7.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней прошло с начала учебного года.
**2**. Добавьте к системному объекту **String** метод **replaceLast** с двумя параметрами. Первый параметр - "что искать", второй - "на что заменить". Ваш метод должен возвращать строку, в контексте которой вызван и в которой последнее вхождение подстроки "что искать" заменено на подстроку "на что заменить".
Напишите программу с тестами для демонстрации работы Вашего метода.
**3**. Возьмите несколько (не менее пяти) любых картинок. Сделайте так, чтобы на Вашей странице отображались одновременно все картинки в различных местах и около каждой картинки был изначально "включенный" чек-бокс. С помощью чек-боксов пользователь должен иметь возможность включать/выключать показ соответствующих картинок. Т.е. немедленно по нажатии чек-бокса картинка должна исчезать (или появляться).
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 8.**

**1**. Создать HTML-страницу, которая при загрузке выводит названия трех случайных карт по одному на строке. Сначала достоинство, а потом масть.

Например:

Тройка бубей

Семерка треф

Дама пик

**2**. Для объекта **String** добавить метод **isDigits()**, который возвращает true, если строка состоит только из цифр, и false - в противном случае.

**3**. Сделайте "осторожный" **reset**. На вашей странице должна быть форма с различными элементами *(не менее пяти)*, такими как поле ввода, чек-бокс, список и т.д. При нажатии кнопки **reset**. Ваша программа - обработчик этого события должна проверять, изменено ли что-либо в форме или все осталось, как было изначально. Если форма изменена, пользователю должен быть задан вопрос (функция **confirm**): "Вы собираетесь сбросить форму. При этом, все, что Вы ввели будет потеряно. Сбросить форму?". Далее форма должна быть сброшена или не сброшена в зависимости от ответа пользователя.

**Вариант 9.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней прошло с 9 мая 1945 года.

**2**. В объект **Array** добавьте метод **notZero()**, который возвращает количество элементов массива, не являющихся нулями.

**3**. Возьмите пять произвольных картинок. Разместите на экране группу кнопок radiobutton. Каждая из кнопок соответствует одной картинке. В зависимости от того, какую опцию выбрал пользователь на экране появляется та или иная картинка.

**Вариант 10.**

**1**. Создать HTML-страницу со случайным цветам текста.

**2**. Написать функцию, которая в строке, переданной в качестве аргумента, заменяет вхождение двух подряд идущих пробелов на один пробел и возвращает полученную строку.

Далее Вам необходимо создать форму с именем "Test", в которой помещены:

1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:

ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"

где *имя_функции* необходимо заменить на имя Вашей функции.

**3**. Сделайте на своей странице будильник. Человек, вошедший на нее, должен иметь возможность указать в некотором текстовом поле желаемое время. В указанное время должно появиться сообщение (**alert**). Если Вам так удобнее, сделайте, чтобы после ввода времени он должен был нажать кнопку "Ввод".

**Вариант 11.**

**1**. Создать HTML-страницу, которая при загрузке выводит таблицу размерами 3x3, каждая клетка которой заполняется случайным образом либо символом "+", либо символом "o".

**2**. Для объекта **Date** добавить метод **MonthName()**, который возвращает строку с названием месяца по-русски ("Январь", "Февраль" и т.д.).

Добавьте к системному объекту **Date** метод **print**. Ваш метод, должен возвращать строку для печати. Строка, будучи напечатана с помощью document.write, должна содержать дату по-русски черным цветом, если **Date** задает рабочий день, и красным - если субботу или воскресенье.

Напишите программу с тестами для демонстрации работы Вашего метода.
**3**. Сделайте страницу с несколькими (не менее пяти) однострочковыми полями для ввода текста. У формы должно обрабатываться событие **submit**. Обработка состоит в следующем: если все поля формы заполнены (не пусты), выдается сообщение (**alert**) "О'кей". Если же хотя бы одно поле пустое, выдается сообщение об ошибке и, как только человек нажмет "Ok", курсор устанавливается в ошибочное поле.
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Вариант 12.**

**1**. Создать HTML-страницу, которая печатает Ваш возраст в годах, месяцах и днях.
**2**. Написать функцию, которая возвращает количество символов 'a' в строке, переданной в качестве аргумента.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1. поля ввода с именем "In" и с именем "Result";
2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Сделайте "навязчивую" ссылку. Это обычная ссылка на другую страницу, но, если пользователь нацелит на нее мышку, а затем, не нажав на ссылку, попытается убрать мышку в сторону, будет выдано сообщение: "Эй!!! А нажать на меня? Забыл? Нажать сейчас?". Это сообщение должно выдаваться функцией **confirm**, чтобы человек мог ответить "Да" или "Нет". Если ответ - "Да", должен быть осуществлен переход так, как если бы он нажал на ссылку. Разместите несколько таких ссылок на странице.

**Вариант 13.**

**1**. Создать HTML-страницу, которая при загрузке выводит названия комбинацию игры в кости - два целых случайных числа от 1 до 6.
**2**. Написать функцию, которая проверяет, состоит ли строка из символов “*”.
Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1. поля ввода с именем "In" и с именем "Result";
2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Добавьте к системному объекту **String** метод **find** с одним параметром – строкой “что найти”. Ваш метод должен возвращать число вхождений параметра в строку, в контексте которой вызван. Напишите программу с тестами для демонстрации работы Вашего метода.

**Вариант 14.**

**1**. Создать HTML-страницу, которая при загрузке сообщает, сколько дней осталось до первого января 2014 года.
**2**. Написать функцию, которая проверяет, что в строке, переданной в качестве аргумента, введена хотя бы одна цифра. Если это так, то функция возвращает *true*, в противном случае - *false*. Далее Вам необходимо создать форму с именем "Test", в которой помещены:
1 поля ввода с именем "In" и с именем "Result";

2. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. В объект **Array** добавьте метод **notEmpty()**, который возвращает количество элементов массива, не являющихся пустыми строками.

**Вариант 15.**

**1**. Создать HTML-страницу, которая выводит коэффициент Ваших физических способностей *F(l)* на ближайший выходной, используя формулу

I=l^3/sin(l),
где *l* - количество дней, прошедших от Вашего рождения.
**2**. Написать функцию, которая определяет пол человека по его имени, отчеству и фамилии.
Для этого Вам необходимо создать форму с именем "Test", в которой помещены:
1. поле ввода с именем "In";
2. поле ввода с именем "Result";
3. кнопка "Enter", которая при нажатии должна вызывать Вашу функцию, передав ей в качестве аргумента строку из поля "In", а результат работы функции вывести в поле "Result". Более конкретно, обработка нажатия кнопки должна выглядеть так:
ONCLICK="document.Test.Result.value=*имя_функции*(document.Test.In.value)"
где *имя_функции* необходимо заменить на имя Вашей функции.
**3**. Сделайте страницу, на которой сначала не видно ни одной картинки, она выглядит пустой. Однако стоит набрать на клавиатуре слово "show", как картинки немедленно появляются. Если потом набрать "hide", они исчезают.
Эту задачу можно решать различными способами, но самый простой - определить необходимые объекты или добавить нужные методы к системным объектам.

**Критерии оценивания (оценочное средство - Контрольная работа)**

| Оценка | Критерии оценивания |
|---|---|
| превосходно | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного и дополнительного материала |
| отлично | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание основного материала |
| очень хорошо | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, обучающийся отвечает четко и последовательно, показывает глубокое знание материала, допущено не более 2 неточностей непринципиального характера |

| Оценка | Критерии оценивания |
|---|---|
| хорошо | Задание выполнено в полном объеме (все поставленные задачи решены), ответ логичен и обоснован, допущены неточности непринципиального характера, но обучающийся показывает систему знаний по теме своими ответами на поставленные вопросы |
| удовлетворительно | Задание выполнено не в полном объеме (решено более 50% поставленных задач), но обучающийся допускает ошибки, нарушена последовательность ответа, но в целом раскрывает содержание основного материала |
| неудовлетворительно | Задание выполнено не в полном объеме (решено менее 50% поставленных задач), обучающийся дает неверную информацию при ответе на поставленные задачи, допускает грубые ошибки при толковании материала, демонстрирует незнание основных терминов и понятий |
| плохо | Задание не выполнено, обучающийся демонстрирует полное незнание материала |

### 5.1.3 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-11:

1.Укажите правильный вариант определения изображения в качестве гиперссылки.
- <a HREF="адрес файла"> IMG SRC="image.gif">
- <a HREF="image.gif">
- <a HREF="адрес файла"><IMG="image.gif">
- <a HRF="адрес файла"><IMG="image.gif">
- <a HREF="адрес файла"><IMG="image">

2. Найдите ошибочное определение гиперссылки.
- <a HREF="alexfine.htm" TARGET="left"> alexfine
- <a TARGET="alexfine.htm" HREF="new"> alexfine
- <a HREF="alexfine.htm"> alexfine
- <a TARG="alexfine.htm" HREF="new"> alexfine
- <a HREF="alexfine"> alexfine

3. В какой таблице ширина промежутков между ячейками составит 20 пикселей?
- <table cellspacing="20">
- <table gridspacing="20">
- <table cellpadding="20">
- <table gridspacing="40">
- <table cellpadding="20p">

4. Как указать выравнивание текста в ячейке таблицы?
- с помощью атрибута CELLPADDING
- с помощью атрибутов VALIGN,ALIGN
- с помощью атрибута gridspacing

- с помощью атрибута cellspacing
- с помощью атрибута gridspace

5. Какой атрибут элемента FORM определяет список кодировок для водимых данных?
- alt
- accept-charset
- enctype-charset
- act-charset
- enct-charset

6. Что определяет атрибут CELLSPACING у элемента разметки TABLE?
- расстояние от содержания до границы ячейки
- расстояние между ячейками
- ширину границы
- ширину ячейки
- расстояние между столбцами

7. Какой атрибут тэга BODY позволяет задать цвет фона страницы?
- color
- background
- set
- bgcolor
- colorofbackground

8. Какой атрибут тега <iMg> задает горизонтальное расстояние между вертикальной границей страницы и изображением?
- BORDER
- HSPACE
- VSPACE
- MSPACE
- GSPACE

9. Какой из приведенных тегов позволяет создавать нумерованные списки?
- OL
- DL
- UL
- DT
- NT

10. Какой полный URL будет сформирован для ссылки в приведенном фрагменте? <bAse href="/"<a">http://alexfine.ru"><BODY><A HREF="doc1.html">Документ 1
- http://alexfine.ru/docs/doc1.html
- http://alexfine.ru/doc1.html
- правильный URL не может быть сформирован
- http://alexfine.ru/users/alexfine/docs/doc1.html
- http://alexfine.ru/users/docs/doc1.html

11. В каких случаях атрибут выравнивания align имеет более высокий приоритет?
- <TH align="left">
- <COL align="left">

- <TABIE align="left">
- <OL align="left">
- <UL align="left">

12. Какой атрибут принадлежит тегу <AREA>?
- SRC
- SHAPE
- CIRCLE
- TABLE
- SRC

13. Какай тэг определяет заголовок документа HTML?
- HTML
- ISINDEX
- BODY
- HEAD
- TITLE

14. Какой из приведенных примеров задает гипертекстовую ссылку из документа 1.html на другой документ?
- <A HREF="#m1">ссылка
- <A HREF=m1>ссылка
- <A HREF="2.html#m1">ссылка
- <A HREF=m2>ссылка
- <A HREF="2.html&&m1">ссылка

15. Выберите вариант корректного описания синтаксиса тега SCRIPT.
- <sCripT Type="тип_языка_программирования">текст программы
- <sCripT nAME="язык_программирования">текст программы
- <sCripT ТУРЕ="тип_документа">текст программы
- <sCripT lang="язык_программирования">текст программы
- <sCripT ТУРЕ="тип_документа"

16. Какой из приведенных фрагментов кода создает переключатель?
- <input Type="checkbox" nAME="a1" vAlue="1"><input ТУРЕ="checkbox" nAME="a1" vAlue="2"><input ТУре="text" nAME="a1" vAlue="2">
- <input ТУРЕ="radiobutton" nAME="a1" vAlue="1"><input ТУРЕ="radiobutton" nAME="a1" vAlue="2">
- <input ТУРЕ="radio" nAME="a1" vAlue="1"><input ТУРЕ="radio" nAME="a1" vAlue="2">
- <input Type="checkbox" nAME="a1" vAlue="1"><input ТУРЕ="checkbox" nAME="a1" vAlue="2"><input ТУре="text" nAME="a1" vAlue="2">
- <input ТУРЕ="radiobutton" nAME="a1" vAlue="1"><input ТУРЕ="radiobutton" nAME="a1" vAlue="2">

17. В какой таблице текст выровнен по центру ячеек?
- <table align=""center"" width=""300"">
- <table align=""left"">
- нет правильного ответа

- <table align=""left"">
- <table align=""right"">

18. Какой тэг определяет тело документа HTML?
- META
- BODY
- HTML
- HEAD
- TITLE

**5.1.4 Типовые задания (оценочное средство - Тест) для оценки сформированности компетенции ПК-9:**

1.Укажите правильный вариант определения изображения в качестве гиперссылки.
- <a HREF="адрес файла"> IMG SRC="image.gif">
- <a HREF="image.gif">
- <a HREF="адрес файла"><IMG="image.gif">
- <a HRF="адрес файла"><IMG="image.gif">
- <a HREF="адрес файла"><IMG="image">

2. Найдите ошибочное определение гиперссылки.
- <a HREF="alexfine.htm" TARGET="left"> alexfine
- <a TARGET="alexfine.htm" HREF="new"> alexfine
- <a HREF="alexfine.htm"> alexfine
- <a TARG="alexfine.htm" HREF="new"> alexfine
- <a HREF="alexfine"> alexfine

3. В какой таблице ширина промежутков между ячейками составит 20 пикселей?
- <table cellspacing="20">
- <table gridspacing="20">
- <table cellpadding="20">
- <table gridspacing="40">
- <table cellpadding="20p">

4. Как указать выравнивание текста в ячейке таблицы?
- с помощью атрибута CELLPADDING
- с помощью атрибутов VALIGN,ALIGN
- с помощью атрибута gridspacing
- с помощью атрибута cellspacing
- с помощью атрибута gridspace

5. Какой атрибут элемента FORM определяет список кодировок для водимых данных?
- alt
- accept-charset
- enctype-charset
- act-charset
- enct-charset

6. Что определяет атрибут CELLSPACING у элемента разметки TABLE?
- расстояние от содержания до границы ячейки

- расстояние между ячейками
- ширину границы
- ширину ячейки
- расстояние между столбцами

7. Какой атрибут тэга BODY позволяет задать цвет фона страницы?
- color
- background
- set
- bgcolor
- colorofbackground

8. Какой атрибут тега <iMg> задает горизонтальное расстояние между вертикальной границей страницы и изображением?
- BORDER
- HSPACE
- VSPACE
- MSPACE
- GSPACE

9. Какой из приведенных тегов позволяет создавать нумерованные списки?
- OL
- DL
- UL
- DT
- NT

10. Какой полный URL будет сформирован для ссылки в приведенном фрагменте? <bAse href="/"<a">http://alexfine.ru"><BODY><A HREF="doc1.html">Документ 1
- http://alexfine.ru/docs/doc1.html
- http://alexfine.ru/doc1.html
- правильный URL не может быть сформирован
- http://alexfine.ru/users/alexfine/docs/doc1.html
- http://alexfine.ru/users/docs/doc1.html

11. В каких случаях атрибут выравнивания align имеет более высокий приоритет?
- <TH align="left">
- <COL align="left">
- <TABIE align="left">
- <OL align="left">
- <UL align="left">

12. Какой атрибут принадлежит тегу <AREA>?
- SRC
- SHAPE
- CIRCLE
- TABLE
- SRC

13. Какай тэг определяет заголовок документа HTML?
- HTML
- ISINDEX
- BODY
- HEAD
- TITLE

14. Какой из приведенных примеров задает гипертекстовую ссылку из документа 1.html на другой документ?
- <A HREF="#m1">ссылка
- <A HREF=m1>ссылка
- <A HREF="2.html#m1">ссылка
- <A HREF=m2>ссылка
- <A HREF="2.html&&m1">ссылка

15. Выберите вариант корректного описания синтаксиса тега SCRIPT.
- <sCripT Type="тип_языка_программирования">текст программы
- <sCripT nAME="язык_программирования">текст программы
- <sCripT ТУРЕ="тип_документа">текст программы
- <sCripT lang="язык_программирования">текст программы
- <sCripT ТУРЕ="тип_документа"

16. Какой из приведенных фрагментов кода создает переключатель?
- <input Type="checkbox" nAME="a1" vAlue="1"><input ТУРЕ="checkbox" nAME="a1" vAlue="2"><input ТУре="text" nAME="a1" vAlue="2">
- <input ТУРЕ="radiobutton" nAME="a1" vAlue="1"><input ТУРЕ="radiobutton" nAME="a1" vAlue="2">
- <input ТУРЕ="radio" nAME="a1" vAlue="1"><input ТУРЕ="radio" nAME="a1" vAlue="2">
- <input Type="checkbox" nAME="a1" vAlue="1"><input ТУРЕ="checkbox" nAME="a1" vAlue="2"><input ТУре="text" nAME="a1" vAlue="2">
- <input ТУРЕ="radiobutton" nAME="a1" vAlue="1"><input ТУРЕ="radiobutton" nAME="a1" vAlue="2">

17. В какой таблице текст выровнен по центру ячеек?
- <table align=""center"" width=""300"">
- <table align=""left"">
- нет правильного ответа
- <table align=""left"">
- <table align=""right"">

18. Какой тэг определяет тело документа HTML?
- META
- BODY
- HTML
- HEAD
- TITLE

**Критерии оценивания (оценочное средство - Тест)**

| Оценка | Критерии оценивания |
|---|---|
| превосходно | 100% правильных ответов |
| отлично | 90-99% правильных ответов |
| очень хорошо | 81-89% правильных ответов |
| хорошо | 66-80% правильных ответов |
| удовлетворительно | 51-65% правильных ответов |
| неудовлетворительно | 31-50% правильных ответов |
| плохо | 30% и меньше правильных ответов |

## 5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

### Шкала оценивания сформированности компетенций

| **Уровень сформированности компетенций (индикатора достижения компетенций)** | **плохо** | **неудовлетворительно** | **удовлетворительно** | **хорошо** | **очень хорошо** | **отлично** | **превосходно** |
|---|---|---|---|---|---|---|---|
| | **не зачтено** | | **зачтено** | | | | |
| Знания | Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа | Уровень знаний ниже минимальных требований. Имели место грубые ошибки | Минимально допустимый уровень знаний. Допущено много негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок | Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет. | Уровень знаний в объеме, превышающем программу подготовки. |
| Умения | Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки | Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном | Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но | Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с | Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетам | Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | объеме | некоторые с недочетами | недочетами. | и, выполнены все задания в полном объеме | |
| Навыки | Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа | При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки | Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами | Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами | Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов | Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов | Продемонстрирован творческий подход к решению нестандартных задач |

## Шкала оценивания при промежуточной аттестации

| Оценка | | Уровень подготовки |
|---|---|---|
| **зачтено** | **превосходно** | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой |
| | **отлично** | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично». |
| | **очень хорошо** | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо» |
| | **хорошо** | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо». |
| | **удовлетворительно** | Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно» |
| **не зачтено** | **неудовлетворительно** | Хотя бы одна компетенция сформирована на уровне «неудовлетворительно». |
| | **плохо** | Хотя бы одна компетенция сформирована на уровне «плохо» |

### 5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

### 5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-11

| |
|---|
| 28. Объект Location. |
| 29. Объект Screen. |

30. Объект Navigator.

31. Объект Event.

32. Способы связи сценариев JavaScript с документом.  Элемент <Script>.

33. Методы расположения сценария в документе.

34. Синтаксис JavaScript . Переменные.

35. Типы данных.

36. Операции JavaScript.

37. Условный оператор.

38. Оператор цикла For.

39. Оператор цикла WHILE.

40. Операторы остановки цикла.

41. Оператор FOR...IN.

42. Оператор WITH.

43. Оператор SWITCH.

44. Функции в JavaScript.

45. Объекты в  JavaScript.

46. Объект GLOBAL.

47. Объект STRING.

48. Объект Date.

49. Объект ARRAY.

50. Объект MATH.

51. Объект BOOLEAN.

52. Объект NUMBER.

53. Объект FUNCTION.

54. Создание объектов.

55. Расширение объектов. Добавление методов к объекту.

**5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции** ПК-9

1. Язык HTML. Понятие разметки, тега, атрибута.

2. Теги структуры.

3. Теги форматирования и оформления.

4. Списки.

5. Таблицы.

6. Работа с графикой.

7. Ссылки.

8. Фреймосодержащие документы.

9. Формы.

10. Элементы формы как объекты.

11. Теги логического разделения. Общие атрибуты тегов.

12. Таблицы стилей. Селекторы.

13. Псевдоклассы и псевдоэлементы.

14. Способы применения таблиц стилей в документе.

15. Свойства таблиц стилей.

16. Позиционирование объектов в документе.

17. Фильтры.

18. Объектная модель.

19. Объект Windows. Свойства объекта Windows.

| |
|---|
| 20. Объект Windows. Методы объекта Windows. |
| 21. Объект Windows.События объекта Window. |
| 22. Объект Document. Свойства объекта Document. |
| 23. Объект Document. Методы объекта Document. |
| 24. Объект Document. События объекта Document. |
| 25. Коллекции объекта Document. Средства работы с коллекциями. |
| 26. Объект Style. |
| 27. Объект History. |

**Критерии оценивания (оценочное средство - Контрольные вопросы)**

| Оценка | Критерии оценивания |
|---|---|
| превосходно | |
| отлично | |
| очень хорошо | |
| хорошо | |
| удовлетворительно | |
| неудовлетворительно | |
| плохо | |

**6. Учебно-методическое и информационное обеспечение дисциплины (модуля)**

Основная литература:

1. Рощин С.М. Современные интернет-технологии. Семь главных трендов : монография / Рощин С.М. - Москва : Дашков и К, 2022. - 124 с. - ISBN 978-5-394-04846-3., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=808223&idb=0.
2. Гуриков Сергей Ростиславович. Интернет-технологии : Учебное пособие / Московский технический университет связи и информатики. - 2. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2023. - 174 с. - ВО - Бакалавриат. - ISBN 978-5-16-016517-2. - ISBN 978-5-16-108029-

0., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=836220&idb=0.
3. Технологии создания интеллектуальных устройств, подключенных к интернет / Приемышев А. В.,Крутов В. Н.,Треяль В. А.,Коршакова О. А. - 2-е изд., стер. - Санкт-Петербург : Лань, 2022. - 100 с. - Книга из коллекции Лань - Инженерно-технические науки. - ISBN 978-5-8114-2310-1., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=800342&idb=0.

Дополнительная литература:

1. Гуриков Сергей Ростиславович. Интернет-технологии : Учебное пособие / Московский технический университет связи и информатики. - 2. - Москва : ООО "Научно-издательский центр ИНФРА-М", 2022. - 174 с. - ВО - Бакалавриат. - ISBN 978-5-16-016517-2. - ISBN 978-5-16-108029-0., https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=792277&idb=0.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Операционная система Microsoft Windows
2. Пакет прикладных программ Microsoft Office
3. Правовая система «Консультант плюс»
4. Правовая система «Гарант».
5. Интернет браузеры (Mozilla Firefox, Google Chrome)
6. Свободно распространяемая среда разработки Pascal ABC.
7. Среда разработки программного обеспечения Lazarus.
8. Свободная интегрированная среда разработки приложений Dev-C++.

**7. Материально-техническое обеспечение дисциплины (модуля)**

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами, специализированным оборудованием: Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой, оснащенные оборудованием и техническими средствами обучения: компьютерная техника с подключением к сети «Интернет», экран, проектор для вывода мультимедиа материалов на экран, динамики для воспроизведения звука, доска.
Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Специальные условия организации обучения по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья

Организация обучения по дисциплине инвалидов и лиц с ограниченными возможностями здоровья осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья при наличии таких обучающихся путем создания специальных условий для получения образования.
Профессорско-преподавательский состав знакомится с психолого-физиологическими особенностями обучающихся инвалидов и лиц с ограниченными возможностями здоровья, индивидуальными программами реабилитации инвалидов (при наличии).
В соответствии с Методическими рекомендациями по организации образовательного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса, утв.

Минобрнауки РФ 08.04.2014 АК-44/05вн при изучении дисциплины предполагается использование социально-активных и рефлексивных методов обучения, технологий социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе.
При освоении дисциплины используются различные сочетания видов учебной работы с методами и формами активизации познавательной деятельности обучающихся для достижения запланированных результатов обучения и формирования компетенций. Форма проведения промежуточной аттестации для обучающихся-инвалидов и лиц с ограниченными возможностями здоровья устанавливается с учетом индивидуальных психофизиологических особенностей. По личной просьбе обучающегося с ограниченными возможностями здоровья, изложенной в форме письменного заявления, по дисциплине предусматриваются:
- замена устного ответа на письменный ответ при сдаче зачёта, экзамена;
- увеличение продолжительности времени на подготовку к ответу на зачёте, экзамене;
- при подведении результатов промежуточной аттестации студентов выставляется максимальное количество баллов за посещаемость аудиторных занятий.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 09.03.03 - Прикладная информатика.

Автор(ы): Васин Дмитрий Юрьевич, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 27.11.2023, протокол № 5.