

Документ подписан простой электронной подписью Информация о владельце: ФИО: Гаскаев Сергей Валерьевич Должность: Ректор Дата подписания: 08.07.2024 09:53:43 Уникальный программный ключ: 89419241880198533607754861093000883783073	МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ Федеральное государственное бюджетное образовательное учреждение высшего образования «Челябинский государственный университет» (ФГБОУ ВО «ЧелГУ»)	Рабочая программа дисциплины "Скриптовые языки программирования" по направлению подготовки (специальности) 01.03.02 "Прикладная математика и информатика" направленности (профилю) Прикладная математика и искусственный интеллект ФГБОУ ВО «ЧелГУ»	стр. 1
---	--	---	--------

Рабочая программа дисциплины (модуля)*

Скриптовые языки программирования

Направление подготовки (специальность)

01.03.02 Прикладная математика и информатика

Направленность (профиль)

Прикладная математика и искусственный интеллект

Присваиваемая квалификация (степень)

бакалавр

Форма обучения

очная

Год набора 2024

*Рабочая программа дисциплины (модуля) адаптирована для инклюзивного обучения инвалидов и лиц с ограниченными возможностями здоровья

Челябинск 2023 г.



Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре ОПОП
3. Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля)
4. Объем дисциплины (модуля)
5. Структура и содержание дисциплины (модуля)
6. Фонд оценочных средств
 - 6.1. Перечень видов оценочных средств
 - 6.2. Типовые контрольные задания и иные материалы для текущей аттестации
 - 6.3. Типовые контрольные вопросы и задания для промежуточной аттестации
 - 6.4. Критерии оценивания
7. Учебно-методическое и информационное обеспечение дисциплины (модуля)
 - 7.1. Рекомендуемая литература
 - 7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"
 - 7.3. Перечень информационных технологий
8. Материально-техническое обеспечение дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Специальные условия освоения дисциплины обучающимися с инвалидностью и ограниченными возможностями здоровья



1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Преподаваемая дисциплина является средством решения специальных прикладных задач. Преподавание и изучение дисциплины следует рассматривать как полезную составляющую профессиональной подготовки. Цель преподавания дисциплины - изучить технологии, обеспечивающие гибкость тиражного ПО, настройку ПО на функциональные требования конкретных пользователей, операционную и аппаратную среду. Задачи дисциплины заключаются в том, чтобы научить студентов технологиям применения скриптовых языков программирования, встраивания их в различные прикладные системы для обеспечения автоматизации и повышения эффективности работы пользователей; использовать скриптовые языки для разработки архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения; изучить интерфейсы библиотек, пакетов программ, продуктов системного и прикладного программного обеспечения и применять их для разработки программного и информационного обеспечения, автоматизации задач операционных систем, разработки сервисов и распределенных информационных систем

Результаты обучения по дисциплине направлены на достижение индикаторов, соответствующих компетенции ПК-1:

ПК-1.1: способы и особенности реализации объектной модели в скриптовых языках программирования.

ПК-1.2: встраивать скриптовые языки в компилируемые среды с целью автоматизации офисных приложений.

ПК-1.3: автоматизацией действий в операционной системе с помощью скриптовых языков программирования.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Цикл (раздел) ОПОП: ФТД.01

2.1 Требования к предварительной подготовке обучающегося:

Знает: [УК-1.1. 3-3.] современное состояние информационно-коммуникационных технологий в мире и перспективы их развития, принципы работы современных информационных технологий, [ОПК-1.1. 3-1.] рынок

информационных систем и информационно-коммуникационных технологий, автоматизирующих организационно-технические и экономические процессы, [ПК-2.1. 3-1.] основные программные платформы и компоненты систем искусственного интеллекта: механизмы логического вывода (рассуждений), объяснений, приобретения знаний, интеллектуальных интерфейсов, принципы Data Ops и Dev Ops, современные компьютерные технологии разработки программных систем

Умеет: использовать современные информационные технологии для решения задач профессиональной деятельности, [ПК-2.1. У-1.] настраивать основные программные платформы и компоненты систем искусственного интеллекта: механизмов логического вывода (рассуждений), объяснений, приобретения знаний, интеллектуальных интерфейсов на особенности проблемной области, участвует в их разработке

Имеет практический опыт: применения современных информационных технологий при проектировании систем искусственного интеллекта, участия в разработке прикладного программного обеспечения

Современные технологии разработки программных систем искусственного интеллекта

2.2 Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

ПК-1: Способен активно участвовать в разработке системного и прикладного программного обеспечения

Знать:

Для достижения ПК-1.1:

способы и особенности реализации объектной модели в скриптовых языках программирования.

Уметь:

Для достижения ПК-1.2:

встраивать скриптовые языки в компилируемые среды с целью автоматизации офисных приложений.

Владеть:

Для достижения ПК-1.3:

автоматизацией действий в операционной системе с помощью скриптовых языков программирования.

В результате освоения дисциплины обучающийся должен

3.1 Знать:



3.1.1	технологии, обеспечивающие гибкость тиражного ПО, настройку ПО на функциональные требования конкретных пользователей, операционную и аппаратную среду;
3.1.2	интерфейсы библиотек, пакетов программ, продуктов системного и прикладного программного обеспечения.
3.2	Уметь:
3.2.1	использовать скриптовые языки для разработки архитектуры, алгоритмических и программных решений системного и прикладного программного обеспечения;
3.2.2	применять интерфейсы библиотек, пакетов программ для разработки программного и информационного обеспечения, автоматизации задач операционных систем, разработки сервисов и распределенных информационных систем.
3.3	Владеть:
3.3.1	применения скриптовых языков программирования, встраивания их в различные прикладные системы для обеспечения автоматизации и повышения эффективности работы пользователей.

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость	4 ЗЕТ
Часов по учебному плану : 144 в том числе : аудиторные занятия : 96 самостоятельная работа : 39,5 : контактная работа: 104,5 ИКР: 8,5	Виды контроля в семестрах: зачеты с оценкой 7

5. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература
	Раздел 1. Применение и реализация скриптовых языков программирования			
1.1	Цели и способы обеспечения гибкости ПО. Использование макросов и скриптов для повышения эффективности работы пользователей. /Лек/	7	2	Л1.1 Л1.2Л2.1 Л2.2 Э1
1.2	Разработка интерпретаторов. Технологии оптимизации исполнения интерпретируемых программ. /Лек/	7	2	Л1.2Л2.2 Э1
1.3	Самомодификация и генерация программ. Метапрограммирование. /Лек/	7	2	Л1.2 Э1
1.4	Самомодификация и генерация программ. Метапрограммирование. /Лек/	7	2	Л1.2 Э1
1.5	Использование интерпретаторов в архитектуре ПО. /Пр/	7	2	Л1.2 Э1
1.6	Разработка простого интерпретатора. /Пр/	7	4	Л1.2 Э1
1.7	Сравнение скорости работы интерпретируемых языков (JavaScript, Python, Lua, PHP) с компилируемым языком (C++) . /Лаб/	7	2	Л1.2 Э1
1.8	Реализация простого интерпретатора для конфигурирования приложения. /Лаб/	7	2	Л1.2 Э1
1.9	Разработка командных файлов .cmd/.ps1. /Лаб/	7	2	Л1.2 Э1
	Раздел 2. Скриптовые языки в ОС, офисных и нативных приложениях			
2.1	Командные файлы. Автоматизация действий в ОС. /Лек/	7	2	Л1.2 Л1.3 Л1.4 Э1



2.2	Типы данных и базовые конструкции интерпретаторов bash, cmd, PowerShell. /Лек/	7	2	Л1.2 Л1.3 Л1.4Л2.2 Э1
2.3	Язык программирования Lua. Место интерпретатора в архитектуре системы. /Лек/	7	2	Л1.3 Э1
2.4	Типы данных и базовые конструкции языка Lua. /Лек/	7	2	Л1.3 Э1
2.5	Использование Lua в проектах на C++ и C#. /Лек/	7	2	Л1.3 Э1
2.6	Языки программирования VBA и StarBasic. Создание макросов и модификация. /Лек/	7	2	Л1.2 Л1.3 Л1.4 Э1
2.7	Автоматизации офисных приложений. Доступ к содержанию текстовых документов, электронных таблиц и презентаций. /Лек/	7	2	Л1.2 Л1.3 Л1.4 Э1
2.8	Доступ к текстовым документам, электронным таблицам из языка C#. /Лек/	7	2	Л1.2 Л1.3 Л1.4 Э1
2.9	Примеры использование командных файлов для автоматизации задач в ОС. /Пр/	7	4	Л1.2 Л1.3 Л1.4 Э1
2.10	Автономное использование интерпретатора языка Lua. /Пр/	7	2	Л1.3 Э1
2.11	Интеграция интерпретатора языка Lua в приложение. Доступ к функциям и структурам данных основного приложения из программы на Lua. /Пр/	7	4	Л1.3 Э1
2.12	Примеры использования VBA или StarBasic для автоматизации обработки документов. /Пр/	7	2	Л1.2 Л1.4 Э1
2.13	Интеграция программы на C# с офисными приложениями. /Пр/	7	2	Л1.2 Л1.4 Э1
2.14	Использование командных файлов для автоматизации задач в ОС. /Лаб/	7	4	Л1.2 Л1.4 Э1
2.15	Написание программы на языке LUA. /Лаб/	7	2	Л1.3 Э1
2.16	Разработка архитектуры приложения с использованием интерпретатора LUA. Подключение интерпретатора LUA к приложению. /Лаб/	7	4	Л1.3 Э1
2.17	Использование VBA или StarBasic для автоматизации обработки документов. /Лаб/	7	2	Л1.2 Л1.4Л2.2 Э1
2.18	Интеграция программы с офисными приложениями. /Лаб/	7	4	Л1.2 Л1.3 Л1.4Л2.2 Э1
Раздел 3. Разработка приложений на Javascript				
3.1	Язык программирования JavaScript. Типы данных и базовые конструкции. /Лек/	7	2	Л1.1 Э1
3.2	Особенности объектной модели в языке JavaScript. Использование функциональной парадигмы при программировании на JavaScript. /Лек/	7	2	Л1.1 Э1
3.3	Библиотеки и фреймворки JavaScript. Разработка web- приложений. /Лек/	7	2	Л1.1Л2.1 Э1
3.4	Серверная платформа Node.js. Взаимодействие с сервером. /Лек/	7	2	Л1.1 Э1
3.5	Базовые конструкции языка JavaScript. /Пр/	7	2	Л1.1 Э1
3.6	Классы и объекты в языке JavaScript. /Пр/	7	2	Л1.1 Э1



3.7	Использование JavaScript в браузерном приложении. /Пр/	7	4	Л1.1Л2.1 Э1
3.8	Клиент-серверные архитектуры. Форматы и способы передачи информации. /Пр/	7	4	Л1.1Л2.1 Э1
3.9	Исследование объектной модели JavaScript. Классы и наследование. /Лаб/	7	2	Л1.1Л2.1 Э1
3.10	Разработка архитектуры web-приложения. Использование JavaScript в браузерном приложении. /Лаб/	7	4	Л1.1Л2.1 Э1
3.11	Разработка архитектуры серверного приложения. Взаимодействие браузерного приложения и сервера. Исследование форматов и способов передачи информации. /Лаб/	7	4	Л1.1Л2.1 Э1
Раздел 4. Самостоятельная работа				
4.1	Подготовка к диф.зачету. /Ср/	7	9	Л1.1 Л1.2 Л1.3 Л1.4 Э1
4.2	Подготовка к проведению лабораторных работ и практических занятий. /Ср/	7	30,5	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Э1
Раздел 5. Иная контактная работа				
5.1	Консультации и промежуточная аттестация. /КурсР/	7	8,5	Л1.1 Л1.2 Л1.3 Л1.4Л2.1 Л2.2 Э1

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

6.1. Перечень видов оценочных средств

Лабораторные работы;
Решение тестов текущего контроля;
Вопросы для подготовки к зачету;

6.2. Типовые контрольные задания и иные материалы для текущей аттестации

Образец теста приведен в приложении.
Лабораторная работа № 1. Сравнение скорости работы интерпретируемых языков (JavaScript, Python, Lua, PHP) с компилируемым языком (C++).
Лабораторная работа № 2. Реализация простого интерпретатора для конфигурирования приложения.
Лабораторная работа № 3. Разработка командных файлов .cmd/.ps1.
Лабораторная работа № 4. Использование командных файлов для автоматизации задач в ОС.
Лабораторная работа № 5. Написание программы на языке LUA.
Лабораторная работа № 6. Разработка архитектуры приложения с использованием интерпретатора LUA. Подключение интерпретатора LUA к приложению.
Лабораторная работа № 7. Использование VBA или StarBasic для автоматизации обработки документов.
Лабораторная работа № 8. Интеграция программы с офисными приложениями.
Лабораторная работа № 9. Исследование объектной модели JavaScript. Классы и наследование.
Лабораторная работа № 10. Разработка архитектуры web-приложения. Использование JavaScript в браузерном приложении.
Лабораторная работа № 1. Разработка архитектуры серверного приложения. Взаимодействие браузерного приложения и сервера. Исследование форматов и способов передачи информации.
Образец лабораторной работы приведен в приложении.

6.3. Типовые контрольные вопросы и задания для промежуточной аттестации

Вопросы к зачету
Темы вопросов по разделу 1:
1 Цели и способы обеспечения гибкости ПО.
2 Использование макросов и скриптов для повышения эффективности работы пользователей.
3 Отличия интерпретаторов от компиляторов.
4 Архитектура интерпретатора
5 Самомодификация и генерация программ.
6 Метапрограммирование.



Темы вопросов по разделу 2:

- 1 Командные файлы. Автоматизация действий в ОС.
- 2 Особенности и возможности bash в Linux
- 3 Особенности и возможности cmd в Windows
- 4 Особенности и возможности PowerShell в Windows
- 5 Язык программирования LUA.
- 6 Место интерпретатора LUA в архитектуре системы.
- 7 Языки программирования VBA и StarBasic. Общая характеристика, применение для автоматизации.
- 8 Доступ к содержанию текстовых документов, электронных таблиц и презентаций.

Темы вопросов по разделу 3:

- 1 Язык программирования JavaScript. Назначение
- 2 Основные типы данных JavaScript.
- 3 Объектная модель в JavaScript.
- 4 Анонимные функции в JavaScript.
- 5 Разработка web-приложений. Архитектура и взаимодействие с DOM.
- 6 Библиотеки и фреймворки для JavaScript
- 7 Способы взаимодействия с сервером

6.4. Критерии оценивания

Учебным планом по данной дисциплине предусмотрен зачет с оценкой.

В течение учебного семестра студенты за каждый вид работы получают баллы. Кроме этого на зачете максимально можно получить 20 баллов. Итоговая оценка складывается из суммы баллов, полученных за работу в семестре и за ответ на зачете. Затем полученная сумма баллов переводится в оценку. При этом допускается получение студентом автоматической оценки только по результатам работы в семестре.

Набранные баллы Оценка

25 – 49	неудовлетворительно
50 – 69	удовлетворительно
70 – 90	хорошо
91 – 100	отлично

Описание распределения баллов за выполнение работ контрольных мероприятий:

Текущий контроль:

Задание 1. (Максимально 10 баллов)

Самостоятельно написано решение на 1 интерпретируемом языке - 3 балла, за использование решения с сайта Rosetta Code оценка снижается на 1 балл, отсутствие решения - 0 баллов;

Найдено и адаптировано решение на 2-м интерпретируемом языке - 2 балла, отсутствие решения - 0 баллов;

Самостоятельно написано решение на C++ - 2 балла, за использование решения с сайта Rosetta Code оценка снижается на 1 балл, отсутствие решения - 0 баллов;

Сравнение 3 языков оформлено в виде таблицы - 3 балла, иначе 0 баллов.

Задание 2. (Максимально 10 баллов)

Реализована основная часть - назначение имени значения - 3 балла, иначе 0 баллов;

Реализовано использование ранее определенных значений - 3 балла, иначе 0 баллов;

Реализованы условные операторы - 3 балла, иначе 0 баллов;

Реализовано включение из других файлов - 1 балл, иначе 0 баллов.

Задание 3. (Максимально 10 баллов)

Код решает поставленную задачу — 4 балла, оценка снижается на 1 балл за каждую ошибку;

Студент может ответить на 3 контрольных вопроса по конструкциям PowerShell и CMD, каждый ответ оценивается от 0 до 2 баллов, 2 балла за полный ответ, 1 балл за частично правильный ответ, 0 баллов за отсутствие ответа или неправильный ответ.

Задание 4. (Максимально 10 баллов)

Есть решение задачи 1 — 5 баллов, оценка снижается на 1 балл за каждую ошибку;

Есть решение задачи 2 — 5 баллов, оценка снижается на 1 балл за каждую ошибку.

Задание 5. (Максимально 10 баллов)



Выполнено подключение библиотеки к проекту — 1 балл;
Создана форма с указанными в задании полями и кнопками — 2 балла, иначе 0 баллов;
Выполнена инициализация интерпретатора Lua — 1 балл;
Выполнена загрузка файла с функциями Lua при старте — 1 балл
Правильно написан код на C# (C++) для подзадачи А – 1 балл;
Правильно написан код на C# (C++) для дифференцированный зачет подзадачи В – 1 балл;
Правильно написан код на Lua для подзадачи А – 1 балл;
Правильно написан код на Lua для подзадачи В – 1 балл;
Выполнена обработка ошибок при вызове функции Lua – 1 балл.

Задание 6. (Максимально 10 баллов)

Выполнена подзадача 1 — 4 балла, оценка снижается на 2 балла за каждую ошибку;
Выполнена подзадача 2 — 6 баллов, оценка снижается на 2 балла за каждую ошибку.

Задание 7. (Максимально 10 баллов)

Выполнено подключение к проекту SDK LibreOffice/сборки MS Office – 2 балла, иначе 0 баллов;
Создана форма с таблицей и кнопкой для экспорта – 2 балла, иначе 0 баллов;
Написан код для экспорта данных таблицы – 2 балла, иначе 0 баллов;
Написан код для формирования итоговых формул в Excel/Calc – 2 балла, иначе 0 баллов;
Написан код для установки ширины столбцов в Excel/Calc – 1 балл;
Написан код для размера (свойств) шрифта в ячейках Excel/Calc – 1 балл.

Задание 8. (Максимально 10 баллов)

Есть решение задачи 1 — 5 баллов, оценка снижается на 1 балл за каждую ошибку;
Есть решение задачи 2 — 5 баллов, оценка снижается на 1 балл за каждую ошибку.

Задание 9. (Максимально 10 баллов)

Написан код HTML – 3 балла, оценка снижается на 1 балл за каждую ошибку;
Написан код CSS - 2 балла, оценка снижается на 1 балл за каждую ошибку;
Написан код JavaScript - 4 балла, оценка снижается на 1 балл за каждую ошибку;
Правильно интегрирован код JavaScript и HTML – 1 балл.

Задание 10. (Максимально 10 баллов)

Написан код HTML для клиента – 2 балла, оценка снижается на 1 балл за каждую ошибку;
Написан код JavaScript для сервера - 3 балла, оценка снижается на 1 балл за каждую ошибку;
Написан код JavaScript для клиента - 3 балла, оценка снижается на 1 балл за каждую ошибку;
Выполнено тестирование — 2 балла, иначе 0 баллов;
Задание выполняется в малой группе 2 человека (при нечетном количестве - 3).

Дифференцированный зачет. (Максимально 10 баллов)

Это контрольное мероприятие проводится в форме собеседования.

Задаются два вопроса по пройденным темам.

В первую очередь предлагаются вопросы по темам, которые были оценены на "неудовлетворительно" по текущему контролю.

Каждый ответ оценивается от 0 до 5 баллов в зависимости от полноты ответа, знания терминов.

Шкала оценивания:

Полный, правильный ответ - 5 баллов

Одна неточность, неправильный термин - 4 балла

Частичный ответ - 3 балла

В ответе есть некоторые правильные определения - 2 балла

Нет ответа - 0 баллов

Оценка ставится как сумма баллов за оба ответа.

Бонус-рейтинг. (Максимально 8 баллов)

Активность на занятиях, посещаемость 100% посещение (допускаются пропуски по уважительной причине) - 3 балла;

85-99% посещение - 2 балла;

Работа у доски 1 балл за решение задачи у доски, но не более 5 баллов за семестр.



7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература

7.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Ресурс
Л1.1	Зудилова Т. В., Буркова М. Л.	Web-программирование JavaScript (http://e.lanbook.com/books/element.php?pl1_id=43561)	Санкт- Петербург : НИУ ИТМО, 2012	ЭБС
Л1.2	Осетрова И. С., Осипов Н. А.	Microsoft Visual Basic for Application (http://e.lanbook.com/books/element.php?pl1_id=43575)	Санкт- Петербург : НИУ ИТМО, 2013	ЭБС
Л1.3	Иерузалимски Р.	Программирование на языке Lua (https://e.lanbook.com/book/90107)	Москва : ДМК Пресс, 2014	ЭБС
Л1.4	Питоньяк Э.	OpenOffice.org pro. Автоматизация работы (http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=1209)	Москва : ДМК Пресс, 2009	ЭБС

7.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Ресурс
Л2.1	Перепелица Ф. А.	Разработка интерактивных сайтов с использованием jQuery (https://e.lanbook.com/book/91556)	Санкт- Петербург : НИУ ИТМО, 2015	ЭБС
Л2.2	Коробко И. В.	PowerShell как средство автоматического администрирования (http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4818)	Москва : ДМК Пресс, 2012	ЭБС

7.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

- Э1 Лань [Электронный ресурс] : электронно-библиотечная система (ЭБС) / издательство Лань. – URL:
<http://e.lanbook.com/>.

7.3 Перечень информационных технологий

7.3.1 Программное обеспечение

LMS Moodle

Python

Visual Studio

LibreOffice

7.3.2 Профессиональные базы данных и информационно-справочные системы

1. Научная электронная библиотека eLIBRARY.RU (<https://elibrary.ru/defaultx.asp?>) eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 – . – URL: <https://elibrary.ru>. – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
2. Электронный каталог научной библиотеки ЧелГУ [Электронный ресурс] : база данных / Челяб. гос. ун-т. – Челябинск, 1992 .

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для реализации дисциплины используются учебные аудитории для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, компьютерные классы для проведения лабораторных работ, а также помещения для самостоятельной работы.

Учебные аудитории укомплектованы специализированной мебелью (подразумевается наличие стандартных рабочих (посадочных) мест) и техническими средствами обучения (переносное и / или стационарное мультимедийное оборудование: экран, ноутбук, проектор).



Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий (мультимедийные презентации по отдельным темам, рисунки, таблицы, схемы и т.д).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с подключением к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Учебным планом предусмотрена самостоятельная работа студентов. Самостоятельная работа проводится с целью углубления знаний по дисциплине и предусматривает:

- проработку теоретического материала по учебникам или конспекту лекций с обязательным разбором приведенных примеров;
- подготовку к лабораторным занятиям;
- подготовку к сдаче зачета.

При планировании времени на самостоятельную работу студентам необходимо предусмотреть регулярное повторение пройденного материала. Теоретический материал, законспектированный на лекциях, необходимо дополнять сведениями из литературных источников, представленных в рабочей программе.

В случае применения при изучении дисциплины электронного обучения, дистанционных образовательных технологий общение обучающихся и преподавателя осуществляется в режиме реального или отложенного времени, при этом используются возможности системы дистанционного обучения Moodle и электронная почта.

Большую часть времени обучающиеся самостоятельно работают с учебно-методическими материалами. Студенты имеют возможность консультироваться с преподавателем по всем вопросам, возникающим в ходе самостоятельной работы, посредством электронной почты, сообщений системы дистанционного обучения Moodle.

Доступ обучающегося к учебным ресурсам в режиме отложенного времени, самостоятельной работы осуществляется через сеть Интернет в удобном для него месте, времени и темпе.

При обучении лиц с ограниченными возможностями здоровья электронное обучение, дистанционные образовательные технологии предусматривают возможность приема-передачи информации в доступных для них формах.

Реализация дисциплины с применением электронного обучения, дистанционных образовательных технологий (далее – ЭО, ДОТ) осуществляется на основании «Положения о реализации основных и дополнительных образовательных программ с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Челябинский государственный университет», «Положения о порядке зачета обучающимся по основным профессиональным образовательным программам высшего образования в ФГБОУ ВО «ЧелГУ» результатов освоения в организациях, осуществляющих образовательную деятельность, учебных предметов, курсов, дисциплин (модулей), практик, дополнительных образовательных программ» посредством электронной информационно-образовательной среды ФГБОУ ВО «ЧелГУ». В исключительных случаях (форс-мажор и т.п.) при реализации образовательной деятельности с применением ЭО, ДОТ могут применять компоненты, не входящие в перечень электронной информационно-образовательной среды.

10. СПЕЦИАЛЬНЫЕ УСЛОВИЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ ОБУЧАЮЩИМИСЯ С ИНВАЛИДНОСТЬЮ И ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Освоение дисциплины инвалидами и лицами с ограниченными возможностями здоровья осуществляется с использованием специальных технических средств и голо информационных технологий, предоставляемых Ресурсным учебно-методическим центром по обучению инвалидов и лиц с ограниченными возможностями здоровья ЧелГУ по запросу обучающегося.

1. Мобильные специальные технические средства для лиц с нарушениями зрения: портативный компьютер с вводом/выводом шрифтом Брайля с синтезатором речи «E1Braille-W14J G2»; ноутбуки с программой экранного доступа NVDA; электронные увеличители для удаленного просмотра; видеоувеличители портативные; тифлоплеер; цифровые диктофоны.

2. Мобильные специальные технические средства для лиц с нарушениями слуха: система свободного звукового поля со встроенной совместимостью с FM-устройствами; радиоклассы «Сонет-PCM» с передатчиком, заушным индуктором и индукционной петлей; система информационная для слабослышащих переносная «Исток» А2 со встроенным плеером – звуковым информатором; документ-камера; программируемые слуховые аппараты индивидуального пользования.

3. Ассистивные информационные технологии: программное обеспечение экранного доступа с синтезом речи NVDA; программы экранного увеличения; программы речевого синтеза для компьютеров и ноутбуков; программы речевого синтеза для мобильных устройств; экранная клавиатура; экранная лупа.



При необходимости для обучающихся с нарушениями зрения на рабочих местах для проведения практических или лабораторных занятий устанавливается специальное программное обеспечение (программа речевой навигации NVDA, речевые синтезаторы, экранные лупы).

В учебные аудитории обеспечивается беспрепятственный доступ для обучающихся инвалидов и обучающихся с ограниченными возможностями здоровья. В каждой аудитории, где обучаются инвалиды и лица с ограниченными возможностями здоровья, предусматривается соответствующее количество мест для обучающихся с учетом нарушений их здоровья.

Для освоения дисциплины инвалидам и лицам с ограниченными возможностями здоровья предоставляется доступ к печатным источникам, имеющимся в научной библиотеке ЧелГУ, с помощью специальных технических средств; доступ к электронным источникам, представленным в форме электронного документа в фонде научной библиотеки ЧелГУ или электронно-библиотечных системах, с помощью специальных технических и программных средств (рабочее место для незрячего пользователя с программным обеспечением экранного доступа с синтезом речи NVDA, рабочее место с компьютерным роллером и клавиатурой Clevy с большими кнопками и с разделяющей клавиши накладкой).

Учебно-методические материалы для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа,
- в форме аудиофайла,
- в печатной форме шрифтом Брайля.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Для инвалидов и лиц с ограниченными возможностями здоровья освоение дисциплины может быть частично или полностью осуществлено с использованием дистанционных образовательных технологий (Moodle, Adobe Connect Pro и пр.).

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья используется индивидуальная работа. Под индивидуальной работой подразумевается две формы взаимодействия с преподавателем: индивидуальная учебная работа (консультации), т.е. дополнительное разъяснение учебного материала и углубленное изучение материала с теми обучающимися, которые в этом заинтересованы, и индивидуальная воспитательная работа. Индивидуальные консультации направлены на индивидуализацию обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или обучающимся с ограниченными возможностями здоровья.

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине обеспечивается выполнение следующих дополнительных требований в зависимости от индивидуальных особенностей, обучающихся:

- а) инструкция по порядку проведения процедуры оценивания предоставляется в доступной форме (устно, в письменной форме, в письменной форме шрифтом Брайля, устно с использованием услуг сурдопереводчика);
- б) доступная форма предоставления заданий оценочных средств (в печатной форме, в печатной форме увеличенным шрифтом, в печатной форме шрифтом Брайля, в форме электронного документа, задания зачитываются ассистентом, задания предоставляются с использованием сурдоперевода);
- в) доступная форма предоставления ответов на задания (письменно на бумаге, набор ответов на компьютере, письменно шрифтом Брайля, с использованием услуг ассистента, устно).

При проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями. Эти средства могут быть предоставлены ЧелГУ или могут использоваться собственные технические средства. При необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на задания, процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья допускается с использованием дистанционных образовательных технологий.

Задание 1

Сравнение скорости работы интерпретируемых языков (JavaScript, Python, Lua, PHP) с компилируемым языком (C++)

Содержание работы

- Выберите знакомый язык программирования, например, Python
- Напишите программу, включающую циклы и работу с массивами, например, сортировку пузырьком на выбранном языке программирования
- Напишите тот же код на языке C++
- Используйте сайт Rosetta Code для поиска кода на еще одном скриптовом языке.
- Сравните время выполнения. Количество выполняемых действий должно быть не менее 10^8 . Для сортировки пузырьком размер массива должен быть 10^4 .

Отчет должен включать код на 3 языках программирования, а также таблицу с результатами работы для всех 3 языков, в которой указывается название языка, время выполнения (абсолютное с точностью 0,01 сек) и отношение к времени выполнения программы на C++

Вывод. Почему есть разница во времени работы между языками?

Критерии оценки задания

- Самостоятельно написано решение на 1 интерпретируемом языке - 3 балла
- Найдено и адаптировано решение на 2-м интерпретируемом языке - 2 балла
- Самостоятельно написано решение на C++ - 2 балла
- Сравнение оформлено в виде таблицы - 3 балла

За использование решений с сайта Rosetta Code оценка снижается на 1 балл

Максимальный балл — 10

Задание 2

Реализация простого интерпретатора для конфигурирования приложения

Вариант 1. SciTE:

имя=значение

значение может включать \$(имя) для использования ранее определенных значений

```
if имя
    определения с отступом
import имя_файла
```

Вариант 2. INI:

```
[Раздел]
имя=значение
имя?условие=значение
!имя_файла
```

! включение файла

значение может включать подстановку: %имя%, %раздел:имя%, %% заменяет знак %

Вариант 3. SASS:

```
$имя: значение
Раздел
    имя: значение
    @if переменная
    @else
@mixin имя_mixin
    имя: значение
Раздел
    @include имя_mixin
```

подстановка: \$имя, \$\$ заменяет знак \$

Включение только из другого раздела

Вариант 4. XML:

```
<раздел1>
<имя1>значение</имя1>
<имя2 if="имя">значение</имя2>
<имя3 value="имя1" />
</раздел1>
<раздел2 if="имя">
</раздел2>
<import resource="имя_файла" />
```

имя в value и в if может быть простым или раздел:имя

Вариант 5. HCL:

```
имя= "значение"
раздел {
    имя1= "значение"
    имя2=имя
    ${if имя}
    ${endif}
}
${import имя_файла}
```

имя справа от = и в if может быть простым или раздел.имя

Содержание работы

Отчет должен включать код на C++ или другом языке программирования для загрузки конфигурационного файла, пример конфигурационного файла(ов) и вывод результатов загрузки. При использовании разделов ключ является парой (раздел, имя)

Критерии оценки задания

- Реализована основная часть имя=значение - 3 балла
- Реализовано использование ранее определенных значений - 3 балла
- Реализованы условия - 3 балла
- Реализовано включение из других файлов или @include в варианте 3 - 1 балл

Максимальный балл — 10

Задание 3

Командные файлы

Реализовать с помощью PowerShell или CMD одну из следующих задач

Варианты заданий

1. Сделать ping компьютеров в пределах 192.168.x.1...255 и вывести список найденных. Здесь x взять из IP-адреса текущего компьютера
2. Скопировать файлы, которые были изменены после заданной даты во всех подпапках заданной папки
3. Вывести суммарный размер файлов в подпапках заданной папки (подпапка - размер)
4. Создать html страницу со всеми картинками в текущей папке и загрузить её в браузере по умолчанию
5. Есть программа sum.exe, которая вводит из input.txt два целых числа A B и выводит в output.txt A+B

Дан набор файлов

input1.txt вход
output1.txt выход
input2.txt
output2.txt

Запустить sum.exe для каждого файла и сравнить с соответствующим выходным вывести номера тестов с ошибками или сообщение ОК

Отчет должен включать код на выбранном языке сценариев

Критерии оценки задания

- Код решает поставленную задачу — 4 балла, оценка снижается на 1 балл за каждую ошибку
- Студент может ответить на 3 контрольных вопроса по конструкциям PowerShell и CMD, каждый ответ оценивается от 0 до 2 баллов, 2 балла за полный ответ, 1 балл за частично правильный ответ, 0 баллов за отсутствие ответа или неправильный ответ

Максимальный балл — 10

Примеры контрольных вопросов

1. Перенаправление ввода-вывода программы
2. Получение текущего пути
3. Изменение текущего пути
4. Получение значения переменной окружения
5. Установка значения переменной окружения
6. Получение списка файлов
7. Копирование файлов
8. Изменение имени файла
9. Удаление файла
10. Создание папки
11. Удаление папки
12. Запуск программы
13. Получение списка запущенных программа
14. Условные операторы
15. Арифметический цикл (от 1 до 100)
16. Цикл по файлам папки
17. Цикл по строкам файла

Задание 4

Основные конструкции языка Lua

Нужно решить две задачи из предложенных

Задача 1 (цикл и ветвление)

Вариант 1

Заданы день и месяц рождения, а также текущие день, месяц и год. Определить, сколько дней осталось до дня рождения. Високосные годы – это те, номер которых делится на 400, а также те, номер которых делится на 4, но не делится на 100.

Ограничения: год от 1920 до 3000, месяц – от 1 до 12, день – от 1 до числа дней в месяце.

В первой строке находятся разделённые пробелами день и месяц рождения, во второй – разделённые пробелами текущие день, месяц и год.

Вывести число дней, оставшихся до дня рождения.

Вариант 2

Ввод содержит одну строку текста длиной не более 100 символов.

Вывести введенную строку в обратном порядке.

Вариант 3

Первая строка ввода содержит два целых числа A и B ($1 \leq A, B \leq 100$).

Вывести числа от A до B включительно, каждое число на отдельной строке.

Вариант 4

Возьмем какое-нибудь натуральное число N . Будем изменять его следующим образом: если число четное, то разделим его на 2, если нечетное, прибавим 1. После нескольких таких изменений мы всегда получаем число 1. Например, из числа 11 получается число 12, затем 6, 3, 4, 2 и, наконец, 1. Таким образом, для получения 1 из 11 нужно проделать 6 изменений. Напишите программу, которая вводит натуральное число N ($1 \leq N \leq 10^9$) и выводит количество изменений данного числа до получения 1.

Задача 2 (структуры данных)

Вариант 1

На компьютере под управлением операционной системы Linux имеется каталог, содержащий N файлов. Пользователю требуется скопировать эти файлы на компьютер, работающий под управлением ОС Windows. К сожалению, файловая система Windows имеет странное свойство. Несмотря на то, что она сохраняет большие и малые буквы в именах файлов, имена, отличающиеся только регистром букв, считаются одинаковыми. Например, файлы с именами ChangeLog, CHANGELOG и changelog при копировании на файловую систему Windows попадут в один и тот же файл.

Чтобы избежать потери данных, предлагается при копировании переименовывать файлы по следующим правилам:

- Файлы копируются в порядке перечисления в исходном каталоге.

- Имена файлов считаются одинаковыми, если они совпадают с точностью до регистра.
- Если при копировании очередного файла выяснилось, что файл с таким именем уже был скопирован, то к имени текущего файла добавляется суффикс "1".
- Если имя, полученное после присоединения суффикса, также уже встречалось, то перебираются суффиксы "2", "3", ..., "10", "11", ... до тех пор, пока не найдётся суффикс, дающий уникальное имя.

Входной файл содержит количество имён N , за которым следует N строк с именами. Имена состоят из латинских букв и цифр и имеют длину от 1 до 255 символов.

Выходной файл должен содержать N строк с модифицированными именами файлов.

Вариант 2

Дан набор различных натуральных чисел. Будем называть число "простым для заданного набора", если число не делится ни на одно из чисел набора, кроме самого себя.

Во входном файле в первой строке содержится целое число N ($1 \leq N \leq 100$) – количество чисел в наборе. Во второй строке файла содержатся N различных целых чисел от 1 до 1000000, разделенных пробелами.

В выходной файл вывести "простые для заданного набора" числа, разделяя числа одним пробелом. Числа выводятся в том порядке, в котором они шли во входном файле.

Вариант 3

В школьной компьютерной сети много источников бесперебойного питания (ИБП), которые должны обеспечить работу компьютеров в случае отключения питания. Поэтому важно знать в каком из ИБП уровень заряда оказался ниже критического. ИБП время от времени сообщают о своем уровне заряда на центральный компьютер. Показания уровня заряда могут меняться из-за ошибок измерения, а также подзарядки или разрядки батарей ИБП.

Напишите программу, которая обрабатывает поступившую информацию от ИБП об уровне их заряда и выводит номер ИБП, у которого M последних показаний уровня заряда оказались меньше или равны критического уровня. Если таких ИБП несколько, то вывести номер того из них, у которого сумма последних M показаний меньше. Если таких ИБП все равно несколько, то вывести ИБП с меньшим номером.

В первой строке входного файла содержится четыре целых числа N , L , M и K , разделенных пробелами – количество ИБП N ($1 \leq N \leq 100$), величина критического уровня заряда L ($1 \leq L < 100$), количество рассматриваемых последних показаний M ($1 \leq M \leq 100$) и количество поступивших показаний от ИБП K ($1 \leq K \leq 100000$). Далее следует K строк, в каждой строке два целых числа, разделенных пробелом – показания от ИБП в порядке поступления, сначала указывается номер ИБП от 1 до N , затем уровень заряда в нем от 0 до 100.

В выходной файл вывести K строк, в каждой строке одно целое число – номер ИБП, который нужно высветить после поступления i -го показания, или 0, если заряд ни одного ИБП не достиг критического уровня.

Вариант 4

По данным координатам (x_i, y_i, z_i) всех N кораблей необходимо вычислить для каждого корабля i величину $\sum_j ((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)$.

Входной файл содержит число N ($2 \leq N \leq 10^5$), за которым следует N троек целых чисел x_i, y_i, z_i ($-10^4 \leq x_i, y_i, z_i \leq 10^4$).

В выходной файл следует вывести N чисел – значение штрафа для каждого корабля.

Отчет должен включать код решенных задач

Критерии оценки задания

- Код решает поставленную задачу — 5 баллов за каждую задачу, оценка снижается на 1 балл за каждую ошибку

Задание 5

Интегрирование приложения и Lua

Для интеграции C# и Lua подключить к проекту с помощью nuget библиотеку NLua.

Создать форму, содержащую 2 или 3 поля, 1 метку, 2 кнопки для выполнения действий а) и б). Инициализацию и загрузку файла с функциями Lua выполнять при старте программы. При нажатии на кнопку только вызывать функции.

При нажатии на соответствующую кнопку вызывается одна из функций на Lua.

- а) Прямое взаимодействие с объектами программы. Функции без аргументов передается поля ввода и метка как глобальные объекты для получения данных и вывода результата
- б) Передача аргументов и возврат результат. Функции передается числа (строки) из полей как аргументы и возвращается результат, который нужно занести в метку.

Для C++ использовать библиотеку Luaaa

Варианты заданий

Вариант 1

3 поля Фамилия Имя Отчество -> Фамилия И.О.
так как может потребоваться вариант И.О. Фамилия

Вариант 2

2 поля координаты x, y -> полярные координаты
можно изменить радианы на градусы

Вариант 3

3 поля длины сторон a,b,c -> площадь треугольника
меняется сообщение о результате в случае "нетреугольника"

Вариант 4

2 поля сумма и валюта (USD, EUR, CNY) -> сумма в рублях
меняется способ получения курса валюты
Первоначальный вариант

```
wget -O out.xml -q http://www.cbr.ru/scripts/XML_daily.asp?date_req=15/03/2021
```

Вариант 5

Расчет гипотенузы по 2 катетам

Отчет включает код на C# (C++) и Lua

Критерии оценки задания

Выполнено подключение библиотеки к проекту — 1 балл.

Создана форма с указанными в задании полями и кнопками — 2 балла, иначе 0 баллов.

Выполнена инициализация интерпретатора Lua— 1 балл
Выполнена загрузка файла с функциями Lua при старте — 1 балл
Правильно написан код на C# (C++) для подзадачи А – 1 балл
Правильно написан код на C# (C++) для подзадачи В – 1 балл
Правильно написан код на Lua для подзадачи А – 1 балл
Правильно написан код на Lua для подзадачи В – 1 балл
Выполнена обработка ошибок при вызове функции Lua – 1 балл

Итого 10 баллов

Задание 6

Скрипты (макросы) в офисных приложениях

Можно использовать Excel или LibreOffice Calc.

Вариант 1

Написать макрос, который для выделенных строк, содержащих информацию о ФИО студента, группе и оценке, выполняет следующие действия:

1. Упорядочить записи по Группа, ФИО студента
2. Удалить строки, у которых оценка 2 или меньше

Вариант 2

Имеется таблица, содержащая информацию о ФИО студента и группе. Написать макрос, который выполняет следующие действия:

1. Для каждой группы студентов завести отдельный лист с названием группы
2. Скопировать ФИО студентов из группы на эти листы, сохранив исходный порядок

Вариант 3

Имеется таблица, содержащая информацию о наименовании товара, количестве и сроке годности: Написать макрос, который выполняет следующие действия:

1. Создать лист "Уценка"
2. Скопировать строки, содержащие информацию о товарах, срок годности которых истекает через неделю, на этот лист

Вариант 4

Имеется таблица, содержащая информацию о работе, сроках выполнения (начало-конец) и списка исполнителей через запятую. Информация упорядочена по дате начала. Написать макрос, который выполняет следующие действия:

1. Создать лист "Итого"
2. Создать на ней таблицу Исполнитель-Загрузка в днях

Пример. Взять список ФИО, на новом листе сделать Ф И.О.

```
Sub test()  
Dim col, row, i, kol  
Dim s As String  
Dim parts() As String  
row = Selection.row  
col = Selection.Column  
kol = Selection.Rows.Count  
Dim sheet As Worksheet  
Dim ak As Worksheet  
Set ak = ActiveSheet  
Set sheet = Sheets.Add(After:=Sheets("Лист1"))
```

```
For i = 1 To kol
  s = ak.Cells(row + i - 1, col).Value
  parts = Split(s, " ")
  sheet.Cells(i, 1).Value = parts(0) + " " + Left(parts(1), 1) + "." + Left(parts(2), 1)
Next
End Sub
```

В качестве отчета высылать файл .xlsx или .ods с макросом

Критерии оценки задания

Выполнена подзадача 1 — 4 балла, оценка снижается на 2 балла за каждую ошибку

Выполнена подзадача 2 — 6 баллов, оценка снижается на 2 балла за каждую ошибку

Итого 10 баллов

Задание 7

Интеграция C# или C++ и офисных приложений

Можно делать интеграцию с Excel или LibreOffice Calc (установить SDK).

Разместить на форме таблицу, содержащую несколько столбцов

Добавить кнопку, отправляющую в Excel/Calc содержимое таблицы и заголовки столбцов.

В строке Итого для числовых полей в Excel/Calc должна быть формула, считающая сумму значений в этом столбце.

В качестве отчета высылать проект C#

Вариант 1

Наименование расхода, дата, сумма

Вариант 2

ФИО, Группа, Оценка (прописью)

Итого как отдельная таблица - количество студентов с неявкой, неудовл., удовл., хорошо, отлично

Вариант 3

Наименование товара, кол-во, цена

Добавляется колонка Сумма (формула), Итого для суммы

Вариант 4

ФИО, Отдел, Зарплата Добавить сумму Итого после каждого отдела

Критерии оценки задания

Выполнено подключение к проекту SDK LibreOffice/сборки MS Office – 2 балла, иначе 0 баллов

Создана форма с таблицей и кнопкой для экспорта – 2 балла, иначе 0 баллов

Написан код для экспорта данных таблицы – 2 балла, иначе 0 баллов

Написан код для формирования итоговых формул в Excel/Calc – 2 балла, иначе 0 баллов

Написан код для установки ширины столбцов, размера шрифта в Excel/Calc – 2 балла, иначе 0 баллов

Итого 10 баллов

Задание 4

Основные конструкции языка JavaScript

Нужно решить две задачи из предложенных

Задача 1 (цикл и ветвление)

Вариант 1

Заданы день и месяц рождения, а также текущие день, месяц и год. Определить, сколько дней осталось до дня рождения. Високосные годы – это те, номер которых делится на 400, а также те, номер которых делится на 4, но не делится на 100.

Ограничения: год от 1920 до 3000, месяц – от 1 до 12, день – от 1 до числа дней в месяце.

В первой строке находятся разделённые пробелами день и месяц рождения, во второй – разделённые пробелами текущие день, месяц и год.

Вывести число дней, оставшихся до дня рождения.

Вариант 2

Ввод содержит одну строку текста длиной не более 100 символов.

Вывести введенную строку в обратном порядке.

Вариант 3

Первая строка ввода содержит два целых числа A и B ($1 \leq A, B \leq 100$).

Вывести числа от A до B включительно, каждое число на отдельной строке.

Вариант 4

Возьмем какое-нибудь натуральное число N . Будем изменять его следующим образом: если число четное, то разделим его на 2, если нечетное, прибавим 1. После нескольких таких изменений мы всегда получаем число 1. Например, из числа 11 получается число 12, затем 6, 3, 4, 2 и, наконец, 1. Таким образом, для получения 1 из 11 нужно проделать 6 изменений.

Напишите программу, которая вводит натуральное число N ($1 \leq N \leq 10^9$) и выводит количество изменений данного числа до получения 1.

Задача 2 (структуры данных)

Вариант 1

На компьютере под управлением операционной системы Linux имеется каталог, содержащий N файлов. Пользователю требуется скопировать эти файлы на компьютер, работающий под управлением ОС Windows. К сожалению, файловая система Windows имеет странное свойство. Несмотря на то, что она сохраняет большие и малые буквы в именах файлов, имена, отличающиеся только регистром букв, считаются одинаковыми. Например, файлы с именами ChangeLog, CHANGELOG и changelog при копировании на файловую систему Windows попадут в один и тот же файл.

Чтобы избежать потери данных, предлагается при копировании переименовывать файлы по следующим правилам:

- Файлы копируются в порядке перечисления в исходном каталоге.

- Имена файлов считаются одинаковыми, если они совпадают с точностью до регистра.
- Если при копировании очередного файла выяснилось, что файл с таким именем уже был скопирован, то к имени текущего файла добавляется суффикс "1".
- Если имя, полученное после присоединения суффикса, также уже встречалось, то перебираются суффиксы "2", "3", ..., "10", "11", ... до тех пор, пока не найдётся суффикс, дающий уникальное имя.

Входной файл содержит количество имён N , за которым следует N строк с именами. Имена состоят из латинских букв и цифр и имеют длину от 1 до 255 символов.

Выходной файл должен содержать N строк с модифицированными именами файлов.

Вариант 2

Дан набор различных натуральных чисел. Будем называть число "простым для заданного набора", если число не делится ни на одно из чисел набора, кроме самого себя.

Во входном файле в первой строке содержится целое число N ($1 \leq N \leq 100$) – количество чисел в наборе. Во второй строке файла содержатся N различных целых чисел от 1 до 1000000, разделенных пробелами.

В выходной файл вывести "простые для заданного набора" числа, разделяя числа одним пробелом. Числа выводятся в том порядке, в котором они шли во входном файле.

Вариант 3

В школьной компьютерной сети много источников бесперебойного питания (ИБП), которые должны обеспечить работу компьютеров в случае отключения питания. Поэтому важно знать в каком из ИБП уровень заряда оказался ниже критического. ИБП время от времени сообщают о своем уровне заряда на центральный компьютер. Показания уровня заряда могут меняться из-за ошибок измерения, а также подзарядки или разрядки батарей ИБП.

Напишите программу, которая обрабатывает поступившую информацию от ИБП об уровне их заряда и выводит номер ИБП, у которого M последних показаний уровня заряда оказались меньше или равны критического уровня. Если таких ИБП несколько, то вывести номер того из них, у которого сумма последних M показаний меньше. Если таких ИБП все равно несколько, то вывести ИБП с меньшим номером.

В первой строке входного файла содержится четыре целых числа N , L , M и K , разделенных пробелами – количество ИБП N ($1 \leq N \leq 100$), величина критического уровня заряда L ($1 \leq L < 100$), количество рассматриваемых последних показаний M ($1 \leq M \leq 100$) и количество поступивших показаний от ИБП K ($1 \leq K \leq 100000$). Далее следует K строк, в каждой строке два целых числа, разделенных пробелом – показания от ИБП в порядке поступления, сначала указывается номер ИБП от 1 до N , затем уровень заряда в нем от 0 до 100.

В выходной файл вывести K строк, в каждой строке одно целое число – номер ИБП, который нужно высветить после поступления i -го показания, или 0, если заряд ни одного ИБП не достиг критического уровня.

Вариант 4

По данным координатам (x_i, y_i, z_i) всех N кораблей необходимо вычислить для каждого корабля i величину $\sum_j ((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2)$.

Входной файл содержит число N ($2 \leq N \leq 10^5$), за которым следует N троек целых чисел x_i, y_i, z_i ($-10^4 \leq x_i, y_i, z_i \leq 10^4$).

В выходной файл следует вывести N чисел – значение штрафа для каждого корабля.

Отчет должен включать код решенных задач

Критерии оценки задания

- Код решает поставленную задачу — 5 баллов за каждую задачу, оценка снижается на 1 балл за каждую ошибку

Задание 9

Взаимодействие JavaScript и DOM

В качестве отчета высылать страницу HTML.

Вариант 1

Используя обработчик событий `keyup`, при вводе текста в `input`, содержимое `input`'а должно выводиться в тег `<p>`, причем чем больше вводится текста в `input`, тем больше становится шрифт в теге `<p>`.

Вариант 2

Используя обработчик событий `mouseover`, при наведении мыши на блок `<div>` показать текст и изменить цвет фона, при повторном наведении на блок – скрывать текст и сбросить цвет. В html задать `div`'у фиксированный `width` и `height`.

Вариант 3

Используя свойство `animation`, написать функцию, которая при нажатии по кнопке, перемещает закрашенный квадрат вправо, потом вниз, а после возвращает его на свое место. При повторном нажатии анимация должна повторяться.

Вариант 4

Имеется три заголовка, с которыми связаны текстовые блоки. Сначала блоки скрыты, т.е. `display:none`. Используя анимацию `slide-up` и `slide-down`, необходимо при нажатии на заголовок отображать его текстовый блок, а остальные скрывать.

Вариант 5

Используя анимацию `fade-in` и `fade-out`, необходимо при нажатии на кнопку отображать окно (`<div>`) с сообщением, а при нажатии на окно, скрывать его. Изначально окно скрыто, т.е. `display:none`.

Вариант 6

Используя свойство `animation`, написать функцию, которая при нажатии по картинке, увеличивает ее размер, а при повторном нажатии - уменьшает.

Критерии оценки задания

Написан код HTML – 3 балла, оценка снижается на 1 балл за каждую ошибку

Написан код CSS - 2 балла, оценка снижается на 1 балл за каждую ошибку

Написан код JavaScript - 4 балла, оценка снижается на 1 балл за каждую ошибку

Правильно интегрирован код JavaScript и HTML – 1 балл

Итого 10 баллов

Задание 10

Использование

Задание выполняется в малой группе 2 (при нечетном количестве 3) человека.

1 человек отвечает за клиентскую часть кода, 1 (2) человек — за серверную. После реализации должна быть выполнена проверка корректности работы в многопользовательской среде.

В качестве отчета высылать страницу HTML и скрипт для браузера.

Вариант 1

Создать игру, в которой люди голосуют против друг друга и за кого проголосовало больше людей, отстраняется из игры. Если у двух человек одинаковое количество голосов, выбывающий выбирается случайно. Предусмотреть голосование против себя самого.

Вариант 2

Игра "угадай слово". Участники должны первыми угадать слово. Открытые в данный момент буквы слова видно всем. Участники отвечают независимо друг от друга. Кто нажал на букву и не угадал, получает штраф – он не может отвечать 30 секунд, кто угадал – плюс очки и так пока кто-то не угадает слово.

Вариант 3

Играют два человека в крестики-нолики. Играют два человека. Игрок, который ходит первым, выбирается случайно. Игровое поле три на три.

Вариант 4

Игра ассоциации. Люди встают в очередь в том порядке, в котором подключились к игре. Программа генерирует случайное слово. Первый человек видит слово и должен написать не само слово, а любую ассоциацию, нажать на кнопку завершить ответ, после чего этот ответ передается на сервер. Далее сервер передает эту ассоциацию второму в очереди, после чего второй должен угадать это слово и написать новую ассоциацию к нему и т.д. История ассоциаций каждого человека должна быть сохранена сервере и показана в конце игры вместе с исходным словом.

Критерии оценки задания

Написан код HTML для клиента – 2 балла, оценка снижается на 1 балл за каждую ошибку
Написан код JavaScript для сервера - 3 балла, оценка снижается на 1 балл за каждую ошибку
Написан код JavaScript для клиента - 3 балла, оценка снижается на 1 балл за каждую ошибку
Выполнено тестирование — 2 балла, иначе 0 баллов.

Итого 10 баллов

Пример: Чат

Каждую секунду получаем последние введенные сообщений.

При отправке сообщения получаем общее количество сообщений.

Код для клиента: public/index.html

```
<html>
<body>
<pre id="chat"></pre>
Сообщение: <input id="newmsg" type="text" size="100"><br>
<button id="submit">Отправить</button>
<div id="message"></div>
</body>
<script>
submit.onclick=async function(){
  let response = await fetch("/addcomment?newmsg="+encodeURIComponent(newmsg.value));
  if (response.ok) {
    let data = await response.json();
    message.innerText=data.message;
  }
};
// каждую секунду получаем сообщения и выводим в chat
setInterval(async function(){
  let response = await fetch("/lastcomments");
  if (response.ok) {
    let data = await response.json();
    chat.innerText=data.join('\n');
  }
},1000);
</script>
</html>
```

Код для сервера: server.js

```
let express = require('express');
let app = express();
app.use(express.static(__dirname + '/public'));
app.use(express.json());
app.use(express.urlencoded({ extended: false }));
let msg=[]; // храним сообщения в массиве
// обработка запросов
app.get('/addcomment', function(req, res) { // добавить сообщение
  msg.push(req.query.newmsg);
  res.send({message:msg.length});
});
app.get('/lastcomments', function(req, res) { // последние 10 сообщений
  res.send(msg.slice(-10));
});

app.listen(3000, function () {
  console.log('Listening on port 3000!');
});
```

Вопросы к зачету

Темы вопросов по разделу 1:

- 1 Цели и способы обеспечения гибкости ПО.
- 2 Использование макросов и скриптов для повышения эффективности работы пользователей.
- 3 Отличия интерпретаторов от компиляторов.
- 4 Архитектура интерпретатора
- 5 Самомодификация и генерация программ.
- 6 Метапрограммирование.

Темы вопросов по разделу 2:

- 1 Командные файлы. Автоматизация действий в ОС.
- 2 Особенности и возможности bash в Linux
- 3 Особенности и возможности cmd в Windows
- 4 Особенности и возможности PowerShell в Windows
- 5 Язык программирования LUA.
- 6 Место интерпретатора LUA в архитектуре системы.
- 7 Языки программирования VBA и StarBasic. Общая характеристика, применение для автоматизации.
- 8 Доступ к содержанию текстовых документов, электронных таблиц и презентаций.

Темы вопросов по разделу 3:

- 1 Язык программирования JavaScript. Назначение
- 2 Основные типы данных JavaScript.
- 3 Объектная модель в JavaScript.
- 4 Анонимные функции в JavaScript.
- 5 Разработка web-приложений. Архитектура и взаимодействие с DOM.
- 6 Библиотеки и фреймворки для JavaScript
- 7 Способы взаимодействия с сервером

