

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Луганский государственный университет имени Владимира Даля»**

**Факультет приборостроения, электротехнических и биотехнических систем
Кафедра «Приборы»**

УТВЕРЖДАЮ

Декан факультета



Тарасенко О.В.

(подпись)

2023 года

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ЯЗЫКИ ПРОГРАММИРОВАНИЯ»**

По направлению подготовки: 12.03.04 – Биотехнические системы и технологии

Профиль подготовки: «Биотехнические и медицинские аппараты и системы»

Луганск 2023

Лист согласования РПУД

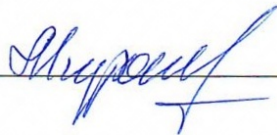
Рабочая программа учебной дисциплины «Языки программирования» по направлению подготовки 12.03.04 Биотехнические системы и технологии -19 с.

Рабочая программа учебной дисциплины «Языки программирования» составлена с учётом Федерального государственного образовательного стандарта высшего образования по направлению подготовки 12.03.04 «Биотехнические системы и технологии» (утверждённый приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. N 950. с изменениями и дополнениями от 26 ноября 2020 г.. 8 февраля 2021 г.)

СОСТАВИТЕЛЬ:

Проф. кафедры «Приборы» Петров А.С.

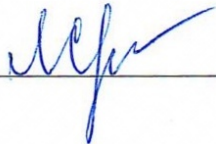
Рабочая программа учебной дисциплины утверждена на заседании кафедры «Приборы» «11» 04 2023 года, протокол № 15

Заведующий кафедрой  Мирошников В.В.

Переутверждена: « » 20 года, протокол №

Согласована (для обеспечивающей кафедры):

Рекомендована на заседании учебно-методической комиссии факультета
Приборостроения электротехнических и биотехнических систем
«18» 04 2023 года, протокол № 3.

Председатель учебно - методической
комиссии факультета приборостроения
электротехнических и биотехнических систем  Яременко С.П.

Структура и содержание дисциплины

1. Цели и задачи дисциплины, её место в учебном процесс

Цель изучения дисциплины – приобретение студентами теоретических и практических знаний о современных технологиях, применяемых в языках программирования разного уровня, в различных оболочках, использование их при разработке управленческих информационных систем и систем целевого назначения, для анализа инженерных решений, для принятия решений в сфере управления предприятием, для создания инструментов программного обеспечения в приборостроении. Это позволит создавать приложения, работающие в разных операционных системах с поддержкой интернет-технологий.

Задачи:

- сформировать у студентов систему знаний о принципах программирования;
- на доступных для понимания примерах, псевдокоде, блок-схемах и других инструментах научить студентов разрабатывать алгоритмы и реализовывать их в программах;
- формирование у студентов представления о методах исследований, которые обеспечивают пошаговый анализ конкретной задачи и демонстрируют обучаемому её решение с целью дальнейшего использования в своей профессиональной деятельности.

2. Место дисциплины в структуре ООП ВО

Дисциплина «Языки программирования» относится к циклу естественно - научных дисциплин (базовая часть).

знания:

основы программирования на языках высокого и низкого уровня, для различных операционных систем, с использованием интернет-технологий;

умения:

применять полученные знания для практической реализации в моделировании процессов, разработке алгоритмов и написании алгоритмов кодов программ;

навыки:

работы с методами и приёмами анализа деятельности человека в различных областях народного хозяйства.

Содержание дисциплины излагается на базе дисциплин основы математики, физика, информатики, вычислительная техника. Является основой для изучения следующих дисциплин: «Моделирование процессов и систем», «Цифровизация инженерной деятельности», «Программное обеспечение измерительной техники», «Базы данных», «Системы передачи данных».

3. Требования к результатам освоения содержания дисциплины

Код и наименование компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.1. Знать принципы работы современных информационных технологий для решения задач профессиональной деятельности ОПК-4.2. Уметь выбирать современные информационные технологии для решения задач профессиональной деятельности ОПК-4.3. Иметь навыки применения современных информационных технологий для решения задач профессиональной деятельности	Знать: принципы работы современных информационных технологий для решения задач профессиональной деятельности. Уметь: выбирать современные информационные технологии для решения задач профессиональной деятельности. Иметь: навыки применения современных информационных технологий для решения задач профессиональной деятельности.
ПК-4. Способен разрабатывать программы и их блоки, проводить их отладку и настройку для решения отдельных задач биотехнических систем	ПК-4.1. Разрабатывает программы и их блоки для решения отдельных задач биотехнических систем ПК-4.2. Проводит отладку и настройку программ для решения отдельных задач биотехнических систем	Знать: основы программирования на языках высокого и низкого уровня, для различных операционных систем, с использованием интернет-технологий. Уметь: Разрабатывает программы и их блоки для решения отдельных задач биотехнических систем. Владеть: Проводит отладку и настройку программ для решения отдельных задач биотехнических систем.

4. Структура и содержание дисциплины

4.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов (9 зач. ед.)					
	Очная форма			Заочная форма		
Общая учебная нагрузка (всего)	90 2,5 з.е	90 2,5 з.е	144 4 з.е.	90 2,5 з.е	90 2,5 з.е	144 4 з.е.
Обязательная аудиторная учебная нагрузка (всего)	51	68	68	8	12	12
в том числе:						
Лекции	34	34	34	6	6	6
Семинарские занятия	-	-	-			
Практические занятия	17	34	34	2	6	6
Лабораторные работы	-	-	-			
Курсовая работа (курсовой проект)	-	-	-			
Другие формы и методы организации образовательного процесса (<i>расчетно-графические работы, групповые дискуссии, ролевые игры, тренинг, компьютерные симуляции, интерактивные лекции, семинары, анализ деловых ситуаций и т.п.</i>)	-	-	-			
Семестр	4	5	6	4	5	6
Самостоятельная работа студента (всего)	39	22	40+36	78+4	74+4	123+9
Форма аттестации	зачет	зачет с оценкой	экзамен	зачет	зачет с оценкой	экзамен

4.2. Содержание разделов дисциплины

Семестр 4

Тема 1. Компьютер: что это такое. Немного истории. Процессор, память, шина. Принципы работы центрального процессора. Внешние устройства. Иерархия запоминающих устройств.

Тема 2. Операционные системы и виды пользовательского интерфейса. История ОС Unix. Unix на домашней машине. Первый сеанс в компьютерном классе. Дерево каталогов. Работа с файлами. Команда и её параметры. Шаблоны имён файлов. История команд и автодописывание имён файлов. Управление выполнением задач. Выполнение в фоновом режиме. Перенаправление потоков ввода-вывода. Редакторы текстов. Права доступа к файлам. Электронная документация (команда man). Командные файлы в Bourne Shell. Переменные окружения. Протоколирование сеанса работы. Графическая подсистема в ОС Unix.

Тема 3. Элементы комбинаторики.

Позиционные системы счисления. Двоичная логика. Виды бесконечности. Алгоритмы и вычислимость. Алгоритм и его свойства.

Тема 4. Программы и данные.

Об измерении количества информации. Машинное представление целых чисел. Числа с плавающей точкой. Тексты и языки. Текст как формат данных. Кодировки. Бинарные и текстовые данные. Машинный код, компиляторы и интерпретаторы.

Тема 5. Язык Паскаль и начала программирования.

Первые программы. Выражения, переменные и операторы. Арифметические операции и понятие типа. Переменные, инициализация и присваивание. Идентификаторы и зарезервированные слова. Ввод информации для её последующей обработки. Берегись нехватки разрядности! Простая последовательность операторов. Конструкция ветвления. Составной оператор. Логические выражения и логический тип. Понятие цикла; оператор while. Цикл с постусловием; оператор goto. Арифметические циклы и оператор for. Вложенные циклы. Побитовые операции. Именованные константы. Разные способы записи чисел.

Тема 6. Подпрограммы. Процедуры. Функции.

Логические функции и условные выражения. Параметры-переменные. Глобальные переменные. Функции и побочные эффекты. Рекурсия.

Тема 7. Конструирование программ.

Концепция структурного программирования. Исключения из правил: операторы выхода. Безусловные переходы. О разбиении программы на подпрограммы.

Тема 8. Символы и их коды; текстовые данные.

Средства работы с символами в Паскале. Посимвольный ввод информации. Чтение до конца файла и программы-фильтры. Чтение чисел до конца файла.

Тема 9. Система типов Паскаля.

Встроенные типы и пользовательские типы. Диапазоны и перечислимые типы. Общее понятие порядкового типа. Массивы. Тип запись. Конструирование сложных структур данных. Пользовательские типы и параметры подпрограмм. Преобразования типов. Строковые литералы и массивы char'ов. Тип string. Встроенные средства работы со строками. Обработка параметров командной строки. Оператор выбора.

Тема 10. Полноэкранные программы.

Вывод в произвольные позиции экрана. Динамический ввод. Управление цветом. Случайные и псевдослучайные числа.

Тема 11. Файлы.

Общие сведения. Текстовые файлы. Типизированные файлы. Блочный ввод-вывод. Операции над файлом как целым. Характеристика и источники загрязнений. Нормирование загрязняющих веществ в водных объектах. Расчеты ПДК в водоемах.

Тема 12. Адреса, указатели и динамическая память.

Что такое указатель. Указатели в Паскале. Динамические переменные. Односвязные списки. Стек и очередь. Проход по списку указателем на указатель. Двусвязные списки; деки. Обзор других динамических структур данных.

Тема 13. Ещё о рекурсии. Взаимная рекурсия.

Ханойские башни. Сопоставление с образцом. Рекурсия при работе со списками. Работа с двоичным деревом поиска.

Тема 14. Ещё об оформлении программ. О роли ASCII-набора и английского языка.

Допустимые стили структурных отступов. Оператор if с веткой else. Особенности оформления оператора выбора. Последовательности взаимоисключающих if'ов. Метки и оператор goto. Максимальная ширина текста программы. Как разбить длинную строку. Пробелы и разделители. Выбор имён (идентификаторов). Регистр букв в именах и ключевых словах. Как справиться с секциями описаний. Непрерывность соблюдения правил.

Тема 15. Тестирование и отладка.

Отладка в жизни программиста. Тесты. Отладочная печать. Отладчик.

Тема 16. Модули и раздельная компиляция.

Модули в Паскале. Использование модулей друг из друга. Модуль как архитектурная единица. Ослабление сцепленности модулей.

Тема 17. Возможности процессора и язык ассемблера.

Классические принципы выполнения программ. Особенности программирования под управлением мультизадачных операционных систем. История платформы i386. Знакомимся с инструментом. Макросы из файла stud_io.inc. Правила оформления ассемблерных программ.

Тема 18. Основы системы команд i386.

Система регистров. Память пользовательской задачи. Сегменты. Директивы для отведения памяти. Команда шопу и виды операндов. Косвенная адресация; исполнительный адрес. Размеры операндов и их допустимые комбинации. Целочисленное сложение и вычитание. Целочисленное умножение и деление. Условные и безусловные переходы. О построении ветвлений и циклов. Условные переходы и регистр ЕСХ; циклы. Побитовые операции. Строковые операции. Ещё несколько интересных команд.

Тема 19. Стек, подпрограммы, рекурсия.

Понятие стека и его предназначение. Организация стека в процессоре i386. Дополнительные команды работы со стеком. Подпрограммы: общие принципы. Вызов подпрограмм и возврат из них. Организация стековых фреймов. Основные конвенции вызовов подпрограмм. Локальные метки. Пример: сопоставление с образцом.

Тема 20. Основные особенности ассемблера NASM.

Ключи и опции командной строки. Основы синтаксиса. Псевдокоманды. Константы. Вычисление выражений во время ассемблирования. Критические выражения.

Тема 21. Макросредства и макропроцессор.

Основные понятия. Простейшие примеры макросов. Однострочные макросы; макропеременные. Условная компиляция. Макроповторения. Многострочные макросы и локальные метки. Макросы с переменным числом параметров. Макродирективы для работы со строками.

Тема 22. Взаимодействие с операционной системой. Мультизадачность и её основные виды.

Аппаратная поддержка мультизадачности. Прерывания и исключения. Системные вызовы и «программные прерывания». Конвенция системных вызовов

ОС LINUX. Конвенция системных вызовов ОС FreeBSD. Примеры системных вызовов. Доступ к параметрам командной строки. Пример: копирование файла.

Тема 23. Раздельная трансляция.

Поддержка модулей в NASM. Пример. Объектный код и машинный код. Библиотеки. Алгоритм работы редактора связей.

Тема 24. Арифметика с плавающей точкой.

Форматы чисел с плавающей точкой. Устройство арифметического сопроцессора. Обмен данными с сопроцессором. Команды арифметических действий. Команды вычисления математических функций. Сравнение и обработка его результатов. Исключительные ситуации и их обработка. Исключения и команда wait. Управление сопроцессором.

Семестр 5

Программирование на Python

Тема 1. Введение в Jupyter Notebook.

Выполнение операторов в Python. Блокноты Jupyter. Блокноты Google Colab. Текстовые ячейки Colab. LaTeX. Ячейки кода Colab. Файлы Colab. Управление документами Colab. Фрагменты кода Colab. Существующие коллекции. Системные псевдонимы. Магические функции. Резюме. Вопросы для закрепления.

Тема 2. Путь разработкиОшибка! Закладка не определена. **Использование примеров кода.**

Что такое программа? Запуск python. Первая программаОшибка! Закладка не определена. Арифметические операторы. Значения и типы. Формальные и естественные языки. ОтладкаОшибка! Закладка не определена. Словарь терминов. Упражнения.

Тема 3. Переменные, выражения и инструкции.

Инструкции присваивания. Имена переменных. Выражения и инструкции. Выполнение скриптов. Приоритет операций. Операции со строками. Комментарии. Отладка. Словарь терминов. Упражнения.

Тема 4. Функции. Вызов функций.

Математические функции. Композиции. Добавление новых функций. Определение и использование. Порядок выполнения. Параметры и аргументы. Переменные и параметры внутри функций — локальные. Стековые диаграммы. Результативные функции и void-функции. Зачем нужны функции? Отладка. Словарь терминов. Упражнения.

Тема 5. Основные типы в Python.

Высокоуровневые и низкоуровневые языки. Операторы. Выполнение базовых математических операций. Классы, объекты и точечная нотация. Резюме. Вопросы для закрепления.

Тема 6. Последовательности.

Общие операции. Проверка принадлежности. Индексирование. Слайсинг. Сбор информации. Математические операции. Списки и кортежи. Создание

списков и кортежей. Добавление и удаление элементов списка. Распаковка. Сортировка списков. Строки. Диапазоны. Резюме. Вопросы для закрепления.

Тема 7. Словари. Множества.

Словари. Создание словарей. Доступ, добавление и обновление помощью ключей. Удаление элементов из словарей. Представления словаря. Проверка наличия ключа в словаре. Метод `get`. Допустимые типы ключей. Метод `hash`. Множества. Операции над множествами. Непересекающиеся множества. Подмножество. Строгие подмножества. Надмножества и строгие надмножества. Объединение. Пересечение. Разность. Симметрическая разность. Обновление множеств. Замороженные множества. Резюме. Вопросы для закрепления.

Тема 8. Управление выполнением.

Составные операторы. Структура составных операторов. Оценка `True` или `False`. Операции проверки на равенство. Операции сравнения. Булевы операции. Оценка объектов. Операторы `if`. Циклы `while`. Циклы `for`. Операторы `break` и `continue`. Резюме. Вопросы для закрепления.

Тема 9. Функции.

Объявление функций. Управляющий оператор. Строки документации. Параметры. Операторы возврата. Область видимости в функциях. Декораторы. Анонимные функции. Резюме. Вопросы для закрепления.

Тема 10. Библиотека NumPy.

Установка и импорт NumPy. Создание массивов. Индексация и слайсинг. Поэлементные операции. Фильтрация значений. Представления и копии. Методы массива. Бродкастинг. Математические функции NumPy. Резюме. Вопросы для закрепления.

Тема 11. Библиотека SciPy.

Обзор SciPy. Подмодуль `scipy.misc`. Подмодуль `scipy.special`. Подмодуль `scipy.stats`. Дискретные распределения. Непрерывные распределения. Резюме. Вопросы для закрепления.

Тема 12. Библиотека Pandas.

Структура датафреймов. Создание датафреймов. Создание датафреймов из словаря. Создание датафреймов из списка списков. Создание датафрейма из файла. Взаимодействие с данными датафреймов. Головы и хвосты. Описательная статистика. Доступ к данным. Синтаксис со скобками. Оптимизированный доступ по метке. Оптимизированный доступ по индексу. Маски и фильтрация. Булевы операторы библиотеки Pandas. Управление датафреймами. Управление данными. Метод `replace`. Интерактивный дисплей. Резюме. Вопросы для закрепления.

Тема 13. Библиотеки визуализации данных.

Библиотека `matplotlib`. Оформление графиков. Маркировка данных. Построение графиков для множества наборов данных. Объектно-ориентированный стиль. Библиотека Seaborn. Темы Seaborn. Библиотека Plotly. Библиотека Bokeh. Другие библиотеки визуализации. Резюме. Вопросы для закрепления.

Тема 14. Библиотеки машинного обучения.

Популярные библиотеки машинного обучения. Принцип работы машинного обучения. Преобразования. Разделение тестовых и тренировочных данных. Обучение и тестирование. Подробнее о Scikit-learn. Резюме. Вопросы для закрепления.

Тема 15. Инструментарий естественного языка (NLTK).

Образцы текстов NLTK. Частотное распределение. Текстовые объекты. Классификация текста. Резюме. Упражнения.

Тема 16. Функциональное программирование.

Знакомство с функциональным программированием. Область видимости и состояние. Зависимость от глобального состояния. Изменение состояния. Изменение изменяемых данных. Функции функционального программирования. Списковые включения. Базовый синтаксис списковых включений. Замена `map` и `filter`. Множественные переменные. Словарные включения. Генераторы. Выражения-генераторы. Функции-генераторы. Резюме. Вопросы для закрепления.

Тема 17. Объектно-ориентированное программирование.

Связывание состояния и функции. Классы и экземпляры. Закрытые методы и переменные. Переменные класса. Специальные методы. Методы представления. Расширенные методы сравнения. Методы математических операторов. Наследование. Резюме. Вопросы для закрепления.

Тема 18. Сортировка. Списки. Чтение и запись файлов.

Контекстные менеджеры. Объекты `datetime`. Регулярные выражения. Наборы символов. Классы символов. Группы. Именованные группы. Найти все. Найти итератор. Замена. Замена с использованием именованных групп. Компиляция регулярных выражений. Резюме. Вопросы для закрепления.

Семестр 6

Системы и сети. Программирование на языке C.

Тема 1. Базовые средства языка C.

Объявления и описания функций. Переменные и их описание. Встроенные типы. Литералы (константы) разных типов. Операции и выражения. Операторы языка C. Перечислимый тип. Введение констант с помощью перечислимого типа. Перечислимый тип и оператор выбора. Локальные «статические» переменные. Адреса, указатели и операции над ними. Организация «выходных» параметров функций. Массивы. Динамическая память. Модификатор `const`. Понятие леводопустимого выражения (*lvalue*). Инициализаторы для массивов. Строки. Строковые литералы. Аргументы командной строки. Точки следования (*sequence points*). Избранные примеры программ.

Тема 2. Стандартные функции ввода-вывода.

Посимвольный ввод-вывод. Форматированный ввод-вывод. Работа с текстовыми файлами. Ввод-вывод отдельных строк. О буферизации ввода-вывода. «Вывод» в строку и «ввод» из строки. Блочный ввод-вывод.

Тема 3. Составной тип данных и динамические структуры.

Структуры. Односвязные списки. Двусвязные списки. Простое бинарное дерево поиска. Объединения и вариантные структуры. Директива typedef.

Тема 4. Макропроцессор.

Предварительные сведения. Макроопределения и макровыводы. Соглашения об именовании. Более сложные возможности макросов. Макросы и конструкция `do { } while(0)`. Директивы условной компиляции. Ещё несколько полезных директив. Директива `#include`. Особенности оформления макродиректив.

Тема 5. Раздельная трансляция.

Общая схема раздельной трансляции в C. Видимость объектов из других модулей. Заголовочные файлы к модулям. Защита от повторного включения.

Объявления типов; неполные типы.

Тема 6. Язык C и стиль кода.

Вспоминаем общие принципы. Фирменные особенности Си. Побочные эффекты.

Возвращаемые значения и выходные параметры. Программы, говорящие по-русски.

Тема 7. Типы указателей.

Многомерные массивы и указатели на массивы. Указатели на функции.

Функции обратного вызова (callback). Сложные описания и общие правила их прочтения.

Тема 8. Ещё о возможностях стандартной библиотеки.

Дополнительные функции работы с кучей. Функции обработки строк.

Генерация псевдослучайных чисел. Средства создания вариативных функций.

Тема 9. Полноэкранные программы на C.

Простой пример. Обработка клавиатурных и других событий. Управление цветом и атрибутами символов. Клавиатурный ввод с тайм-аутами. Обзор остальных возможностей ncurses. Программа на C без стандартной библиотеки.

Тема 10. Объекты и услуги операционной системы.

Операционная система: что это и зачем. Роль и место операционной системы. Управление пользовательскими задачами. Управление внешними устройствами. Границы зоны ответственности ОС. Unix: семейство систем и образ мышления. Замечания о системе X Window. Системные вызовы и их обёртки. О разграничении полномочий.

Тема 11. Ввод-вывод и файловые системы.

Файловая система. Права доступа к файлам. Чтение и запись содержимого файлов. Управление объектами файловой системы. Файлы устройств и классификация устройств. Работа с содержимым каталогов. Отображение файлов в память.

Тема 12. Процессы.

Процесс: что это такое. Свойства процесса. Виртуальное адресное пространство. Порождение процесса. Замена выполняемой программы. Завершение процесса. Ожидание завершения; процессы-зомби. Пример запуска внешней программы. Выполнение процессов и время. Перенаправление потоков ввода-вывода. Полномочия процесса. Количественные ограничения. Обзор

средств взаимодействия процессов. Сигналы. Каналы. Краткие сведения о трассировке.

Тема 13. Терминал и сеанс работы.

Драйвер терминала и дисциплина линии. Сеансы и группы процессов. Управление драйвером терминала. По ту сторону псевдотерминала. Процессы-демоны.

Тема 14. Сети и протоколы.

Сети и сетевые соединения. Шлюзы и маршрутизация. Сетевые протоколы. Понятие протокола и модель OSI. Физические и канальные протоколы. Протокол IP. Дейтаграммы и потоки. Протоколы прикладного слоя. Доменные имена.

Тема 15. Система сокетов в ОС Unix.

Семейства адресации и типы взаимодействия. Сокет и его сетевой адрес.

Тема 16. Приём и передача дейтаграмм.

Потоковые сокеты. Клиент-серверная модель. О «залипании» TCP - порта. Сокеты для связи родственных процессов.

Тема 17. Проблема очерёдности действий и её решения.

Суть проблемы. Решение на основе обслуживающих процессов. Событийно-управляемое программирование. Выборка событий в ОС Unix: вызов select. Сеанс работы как конечный автомат. Неблокирующее установление соединения. Сигналы в роли событий; вызов pselect.

Тема 18. Параллельные программы и разделяемые данные.

О работе с разделяемыми данными. Устаревание и целостность данных. Взаимоисключения. Критические секции. Взаимоисключение с помощью переменных. Понятие мьютекса. О реализации мьютексов. Семафоры Дейкстры.

Тема 19. Классические задачи взаимного исключения.

Задача производителей и потребителей. Задача о пяти философях и проблема тупиков. Граф ожидания. Проблема читателей и писателей. Задача о спящем парикмахере.

Тема 20. Многопоточное программирование в ОС Unix.

Библиотека Posix81x Threads. Семафоры и мьютексы. Демонстрационный пример.

Тема 21. Разделяемые данные на диске.

Обзор имеющихся возможностей. Создание дополнительного файла. Системный вызов flock. Файловые захваты POSIX. Некоторые проблемы файловых захватов.

Тема 22. Ядро операционной системы UNIX.

Основные принципы работы ОС. Ядро как обработчик запросов. Загрузка и жизненный цикл ОС UNIX. Эмуляция физического компьютера. Структура и основные подсистемы ядра. Управление процессами. Процесс как объект ядра системы. Планирование времени процессора. Обработка сигналов.

Тема 23. Управление оперативной памятью.

Проблемы, решаемые менеджером памяти. Виртуальная память и подкачка. Простейшая модель виртуальной памяти. Сегментная организация памяти. Страничная организация памяти. Сегментно-страничная организация памяти.

Тема 24. Управление аппаратурой; ввод-вывод.

Две точки зрения на ввод-вывод. Взаимодействие ОС с аппаратурой. Драйверы. Ввод-вывод на разных уровнях ВС. О роли аппаратных прерываний. Буферизация ввода-вывода. Планирование дисковых обменов. Виртуальная файловая система. Файловая система на диске. Шина, кеш и DMA.

Тема 25. Компилятор gcc. Средства отладки.

Автоматическая сборка: утилита make. Редактор vim: больше возможностей.

Семестр 4

4.3. Лекции

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Управление выполнением задач в ОС Linux.	2	4
2	Программы и данные.	2	
3	Язык Паскаль и начала программирования.	2	
4	Подпрограммы. Процедуры. Функции.	2	
5	Конструирование программ.	2	
6	Система типов Паскаля.	2	2
7	Файлы. Общие сведения. Текстовые файлы.	2	
8	Адреса, указатели и динамическая память. .	2	
9	Тестирование и отладка.	2	
10	Модули и раздельная компиляция. Модули в Паскале.	2	
11	Возможности процессора и язык ассемблера.	2	
12	Основы системы команд i386.	2	
13	Стек, подпрограммы, рекурсия.	2	
14	Основные особенности ассемблера NASM.	2	2
15	Макросредства и макропроцессор.	2	
16	Мультизадачность и её основные виды.	2	
17	Раздельная трансляция. Поддержка модулей в NASM.	2	
	ИТОГО:	34	8

4.4. Практические (семинарские) занятия

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Управление выполнением задач в ОС Linux.	1	4
2	Программы и данные.	1	
3	Язык Паскаль и начала программирования.	1	
4	Подпрограммы. Процедуры. Функции.	1	
5	Конструирование программ.	1	
6	Система типов Паскаля.	1	2
7	Файлы. Общие сведения. Текстовые файлы.	1	
8	Адреса, указатели и динамическая память. .	1	

9	Тестирование и отладка.	1	
10	Модули и раздельная компиляция. Модули в Паскале.	1	
11	Возможности процессора и язык ассемблера.	1	
12	Основы системы команд i386.	1	
13	Стек, подпрограммы, рекурсия.	1	
14	Основные особенности ассемблера NASM.	1	2
15	Макросредства и макропроцессор.	1	
16	Мультизадачность и её основные виды.	1	
17	Раздельная трансляция. Поддержка модулей в NASM.	1	
	ИТОГО:	17	8

4.5. Лабораторные работы Планом не предусмотрены

4.6. Самостоятельная работа студентов

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Управление выполнением задач в ОС Linux.	3	
2	Программы и данные.	2	
3	Язык Паскаль и начала программирования.	2	4
4	Подпрограммы. Процедуры. Функции.	3	
5	Конструирование программ.	2	
6	Система типов Паскаля.	2	
7	Файлы. Общие сведения. Текстовые файлы.	2	
8	Адреса, указатели и динамическая память. .	3	2
9	Тестирование и отладка.	2	
10	Модули и раздельная компиляция. Модули в Паскале.	2	
11	Возможности процессора и язык ассемблера.	2	
12	Основы системы команд i386.	3	
13	Стек, подпрограммы, рекурсия.	2	
14	Основные особенности ассемблера NASM.	3	2
15	Макросредства и макропроцессор.	2	
16	Мультизадачность и её основные виды.	2	
17	Раздельная трансляция. Поддержка модулей в NASM.	2	
	ИТОГО:	39	8

Семестр 5

4.3. Лекции

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Введение в Jupyter Notebook.	2	
2	Запуск python. Первая программа.	2	
3	Переменные, выражения и инструкции.	2	4
4	Функции. Вызов функции. Математические функции.	2	

5	Основы Python. Основные типы в Python.	2	
6	Последовательности. Индексирование. Слайсинг.	2	2
7	Словари. Множества.	2	
8	Функции. Объявление функций.	2	
9	Библиотека NumPy.	2	
10	Библиотека SciPy.	2	
11	Библиотека Pandas.	2	
12	Библиотеки визуализации данных.	2	
13	Библиотеки машинного обучения.	2	
14	Инструментарий естественного языка (NLTK)	2	2
15	Функциональное программирование.	2	
16	Объектно-ориентированное программирование.	2	
17	Сортировка. Списки. Чтение и запись файлов	2	
	ИТОГО:	34	8

4.4. Практические (семинарские) занятия

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Введение в Jupyter Notebook.	2	4
2	Запуск python. Первая программа Ошибка! Закладка не определена.	2	
3	Переменные, выражения и инструкции.	2	
4	Функции. Вызов функции. Математические функции.	2	
5	Основы Python. Основные типы в Python.	2	
6	Последовательности. Индексирование. Слайсинг.	2	2
7	Словари. Множества.	2	
8	Функции. Объявление функций.	2	
9	Библиотека NumPy.	2	
10	Библиотека SciPy.	2	
11	Библиотека Pandas.	2	
12	Библиотеки визуализации данных.	2	
13	Библиотеки машинного обучения.	2	
14	Инструментарий естественного языка (NLTK)	2	2
15	Функциональное программирование.	2	
16	Объектно-ориентированное программирование.	2	
17	Сортировка. Списки. Чтение и запись файлов	2	
	ИТОГО:	34	8

4.5. Лабораторные работы Планом не предусмотрены

4.6. Самостоятельная работа студентов

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Введение в Jupyter Notebook.	1	4
2	Запуск python. Первая программа Ошибка! Закладка не определена.	1	
3	Переменные, выражения и инструкции.	1	
4	Функции. Вызов функции. Математические функции.	1	
5	Основы Python. Основные типы в Python.	1	
6	Последовательности. Индексирование. Слайсинг.	1	2
7	Словари. Множества.	1	
8	Функции. Объявление функций.	1	
9	Библиотека NumPy.	1	
10	Библиотека SciPy.	1	
11	Библиотека Pandas.	1	
12	Библиотеки визуализации данных.	1	
13	Библиотеки машинного обучения.	2	
14	Инструментарий естественного языка (NLTK)	2	2
15	Функциональное программирование.	2	
16	Объектно-ориентированное программирование.	2	
17	Сортировка. Списки. Чтение и запись файлов	2	
	ИТОГО:	22	8

Семестр 6

4.3. Лекции

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Базовые средства языка C. Объявления и описания функций	1	4
2	Стандартные функции ввода-вывода.	1	
3	Составной тип данных и динамические структуры.	1	
4	Макропроцессор.	1	
5	Раздельная трансляция.	2	
6	Язык C и стиль кода. Типы указателей.	2	2
7	Полноэкранные программы на C.	2	
8	Роль и место операционной системы.	2	
9	Ввод-вывод и файловые системы. Права доступа к файлам.	2	
10	Процессы. Свойства процесса.	2	
11	Терминал и сеанс работы.	2	
12	Сети и сетевые соединения. Шлюзы и маршрутизация.	2	
13	Система сокетов в ОС Unix. Семейства адресации.	2	
14	Приём и передача дейтаграмм. Поточковые сокет.	2	2
15	Параллельные программы и разделяемые данные.	2	
16	Многопоточное программирование в ОС Unix.	2	
17	Разделяемые данные на диске.	2	

18	Основные принципы работы ОС. Ядро как обработчик запросов.	1	
19	Управление оперативной памятью.	1	
20	Управление аппаратурой; ввод-вывод.	1	
21	Компилятор gcc. Средства отладки. Автоматическая сборка.	1	
	ИТОГО:	34	8

4.4. Практические (семинарские) занятия

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Базовые средства языка C. Объявления и описания функций	2	4
2	Стандартные функции ввода-вывода.	2	
3	Составной тип данных и динамические структуры.	2	
4	Макропроцессор.	2	
5	Раздельная трансляция.	2	
6	Язык C и стиль кода. Типы указателей.	2	2
7	Полноэкранные программы на C.	2	
8	Роль и место операционной системы.	2	
9	Ввод-вывод и файловые системы. Права доступа к файлам.	2	
10	Процессы. Свойства процесса.	2	
11	Терминал и сеанс работы.	2	
12	Сети и сетевые соединения. Шлюзы и маршрутизация.	2	
13	Система сокетов в ОС Unix. Семейства адресации.	1	
14	Приём и передача дейтаграмм. Потокосокеты.	1	2
15	Параллельные программы и разделяемые данные.	1	
16	Классические задачи взаимного исключения.	1	
17	Многопоточное программирование в ОС Unix.	1	
18	Разделяемые данные на диске.	1	
19	Основные принципы работы ОС. Ядро как обработчик запросов.	1	
20	Управление оперативной памятью.	1	
21	Управление аппаратурой; ввод-вывод.	1	
22	Компилятор gcc. Средства отладки. Автоматическая сборка.	1	
	ИТОГО:	34	8

4.5. Лабораторные работы Планом не предусмотрены

4.6. Самостоятельная работа студентов

№	Название темы	Объем часов	
		Очная форма	Заочная форма
1	Базовые средства языка C. Объявления и описания функций	3	4
2	Стандартные функции ввода-вывода.	3	
3	Составной тип данных и динамические структуры.	3	
4	Макропроцессор.	3	
5	Раздельная трансляция.	3	
6	Язык C и стиль кода. Типы указателей.	3	2

7	Полноэкранные программы на С.	3	
8	Роль и место операционной системы.	3	
9	Ввод-вывод и файловые системы. Права доступа к файлам.	3	
10	Процессы. Свойства процесса.	3	
11	Терминал и сеанс работы.	3	
12	Сети и сетевые соединения. Шлюзы и маршрутизация.	3	
13	Система сокетов в ОС Unix. Семейства адресации.	4	
14	Приём и передача дейтаграмм. Потокосокеты.	4	2
15	Параллельные программы и разделяемые данные.	4	
16	Классические задачи взаимного исключения.	4	
17	Многопоточное программирование в ОС Unix.	4	
18	Разделяемые данные на диске.	4	
19	Основные принципы работы ОС. Ядро как обработчик запросов.	4	
20	Управление оперативной памятью.	4	
21	Управление аппаратурой; ввод-вывод.	4	
22	Компилятор gcc. Средства отладки. Автоматическая сборка.	4	
	ИТОГО:	76	8

4.7. Курсовые работы/проекты учебным планом не предусмотрены.

5. Образовательные технологии

Преподавание дисциплины ведётся с применением следующих видов образовательных технологий:

Информационные технологии: использование электронных образовательных ресурсов (электронные конспект лекций, методические указания к лабораторным работам, методические указания к выполнению расчётно-графических работ, методические указания к самостоятельному изучению дисциплины, размещённые во внутренней сети и сайте кафедры) при подготовке к лекциям и лабораторным занятиям.

Работа в команде: каждая лабораторная работа выполняется несколькими студентами совместно в бригадах по 4-5 чел. Для каждой бригады имеется свое задание, общее для студентов этой бригады. Кроме этого каждый студент получает свое индивидуальное задание к практической работе, частично связанное с вариантом задания к расчётно-графической работе, что позволяет мотивировать каждого студента на совместную работу в команде.

6. Формы контроля освоения дисциплины.

Текущая аттестация студентов производится в дискретные временные интервалы лектором и преподавателем, ведущими практические работы по дисциплине в следующих формах:

- вопросы к практическим работам;
- вопросы к экзамену.

Форма аттестации по результатам освоения дисциплины проходит в форме экзамена (включает в себя ответ на теоретические вопросы и решение задачи). Студентам, выполнившим 75% текущих и контрольных мероприятий на «отлично», а остальные 25 % на «хорошо», имеют право на получение итоговой отличной оценки.

В экзаменационную ведомость и зачётную книжку выставляются оценки по шкале, приведённой в таблице:

Шкала оценивания	Характеристика знания предмета и ответов
отлично (5)	Студент глубоко и в полном объёме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендуемую литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки, непоследовательность в излагаемых ответах в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

7. Учебно-методическое и программно-информационное обеспечение дисциплины

а) основная литература:

1. Столяров А. В. Программирование: введение в профессию - Том 1. Азы программирования : учебное пособие / А. В. Столяров. - Москва : изд-во МАКС ПРЕСС, 2021. - 704 с. <http://www.stolyarov.info>
ISBN 978-5-3170-6573-7
2. Столяров А. В. Программирование: введение в профессию - Том 2. Системы и сети: учебное пособие / А. В. Столяров. - Москва : изд-во МАКС ПРЕСС, 2021. - 704 с. <http://www.stolyarov.info>
ISBN 978-5-3170-6573-7

3. Столяров А. В. Программирование: введение в профессию - Том 3. Парадигмы: учебное пособие / А. В. Столяров. - Москва : изд-во МАКС ПРЕСС, 2021. - 704 с. <http://www.stolyarov.info>
ISBN 978-5-3170-6576-8
4. Марк Лутц. Программирование на PYTHON: Том 1, издание четвертое. Изд-во O'REILLY, Санкт-Петербург, Москва, 2011. - 992 с.
ISBN 978-5-9328-6210-0
5. Марк Лутц. Программирование на PYTHON: Том 2, издание четвертое. Изд-во O'REILLY, Санкт-Петербург, Москва, 2011. - 993 с.
ISBN 978-5-9328-6211-7
6. Н. Секунов. Программирование на C++ в LINUX/. - Санкт-Петербург, «БХВ-Петербург», 2004, 363 с. ISBN 5-94157-355-3

б) дополнительная литература:

1. Тони Геддис. Начинаем программировать на PYTHON: Учебное пособие. Издание пятое – Санкт-Петербург: изд-во БХВ - ПЕТЕРБУРГ, 2022. – 880 с., ил. ISBN 978-5-9775-6803-6
2. Билл Любанович. Простой PYTHON. Современный стиль программирования. Изд-во O'REILLY, – Санкт-Петербург, Москва, 2016. – 480 с. ISBN 978-5-496-02088-6
3. Уэс Маккини. PYTHON и анализ данных: – Изд-во O'REILLY / Перевод с англ. Слипкин А.А. - М.ДМК Пресс, 2015. – 482 с.: ил. ISBN 978-5-97060-315-4
4. Р. Седжвик, К. Уэйн, Р. Дондеро. Программирование на языке PYTHON. Учебный курс. – изд-во Диалектика, Москва, Санкт-Петербург, 2017. – 736 с. ISBN 978-5-9908462-1-0
5. Oleksandr PETROV. Introduction to data mining / Oleksandr PETROV, Oleksandr Shumeyko, Beata BASIURA, Anton Petrov. — Kraków : Wydawnictwa AGH, 2019. — 362, [1] s. — Bibliogr. s. 347–[363]. — ISBN: 978-83-66364-25-7
6. Н. Елманова, С. Трепалин, А. Тенцер. Delphi 6 и технология COM. - Санкт-Петербург, Москва, «Питер», 2002, 640 с. ISBN 5-94723-149-2
7. Д. Гончаров, Т. Салихов. DirectX 7.0. - Учебный курс. Программирование графики.- Санкт-Петербург, Москва, «Питер», 2001, 520 с. ISBN 5-318-00173-4
8. Каранджит С. Сайан. Visual J++. Методология профессионального программирования для Интернет с использованием Visual J++/ изд-во «DiaSoft», Киев, 1997. 528 с. ISBN 966-7033-27-9

в) интернет-ресурсы:

1. Министерство образования и науки Российской Федерации – <http://минобрнауки.рф/>
2. Федеральная служба по надзору в сфере образования и науки – <http://obrnadzor.gov.ru/>
3. Министерство образования и науки Луганской Народной Республики – <https://minobr.su>
4. Народный совет Луганской Народной Республики – <https://nslnr.su>

5. Портал Федеральных государственных образовательных стандартов высшего образования – <http://fgosvo.ru>

6. Федеральный портал «Российское образование» – <http://www.edu.ru/>

7. Информационная система «Единое окно доступа к образовательным ресурсам» – <http://window.edu.ru/>

8. Федеральный центр информационно-образовательных ресурсов – <http://fcior.edu.ru/>

Электронные библиотечные системы и ресурсы

9. Электронно-библиотечная система «Консультант студента» – <http://www.studentlibrary.ru/cgi-bin/mb4x>

10. Электронно-библиотечная система «StudMed.ru» – <https://www.studmed.ru>

Информационный ресурс библиотеки образовательной организации

11. Научная библиотека имени А. Н. Коняева – <http://biblio.dahluniver.ru/>

8. Материально-техническое обеспечение дисциплины

Освоение дисциплины «Языки программирования» предполагает использование академических аудиторий, соответствующих действующим санитарным и противопожарным правилам и нормам.

Лекционные занятия: аудитория, оснащённая презентационной техникой (проектор, экран, компьютер).

Практические занятия: комплект электронных слайдов, аудитория с презентационной техникой (проектор, экран, ПК), компьютерный класс с доступом в Интернет.

Прочее: рабочее место преподавателя, оснащённое компьютером с доступом в Интернет.

Программное обеспечение:

Функциональное назначение	Бесплатное программное обеспечение	Ссылки
Офисный пакет	Libre Office 6.3.1	https://www.libreoffice.org/ https://ru.wikipedia.org/wiki/LibreOffice
Операционная система	UBUNTU 19.04	https://ubuntu.com/ https://ru.wikipedia.org/wiki/Ubuntu
Браузер	Firefox Mozilla	http://www.mozilla.org/ru/firefox/fx
Браузер	Opera	http://www.opera.com
Почтовый клиент	Mozilla Thunderbird	http://www.mozilla.org/ru/thunderbird
Файл-менеджер	Far Manager	http://www.farmanager.com/download.php

Архиватор	7Zip	http://www.7-zip.org/
Графический редактор	GIMP (GNU Image Manipulation Program)	http://www.gimp.org/ http://gimp.ru/viewpage.php?page_id=8 http://ru.wikipedia.org/wiki/GIMP
Программа трехмерного проектирования	КОМПАС-3D v18.1	https://kompas.ru/kompas-3d/download/
Редактор PDF	PDFCreator	http://www.pdfforge.org/pdfcreator

9. Оценочные средства по дисциплине

Паспорт фонда оценочных средств по учебной дисциплине «Языки программирования»

Перечень компетенций (элементов компетенций), формируемых в результате освоения учебной дисциплины (модуля) или практики

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Контролируемые темы учебной дисциплины, практики	Этапы формирования (семестр изучения)
1	ОПК-4	Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.1. Знать принципы работы современных информационных технологий для решения задач профессиональной деятельности ОПК-4.2. Уметь выбирать современные информационные технологии для решения задач профессиональной деятельности ОПК-4.3. Иметь навыки применения современных информационных технологий	Тема 2. Операционные системы и виды пользовательского интерфейса.	4
				Тема 5. Язык Паскаль и начала программирования.	4
				Тема 6. Подпрограммы. Процедуры. Функции.	4
				Тема 12. Адреса, указатели и динамическая память.	4
				Тема 17. Возможности процессора и язык ассемблера.	4
				Тема 20. Основные особенности ассемблера NASM.	4

			технологий для решения задач профессиональной деятельности	Тема 2. Путь разработки и использование примеров кода.	5
				Тема 4. Функции. Вызов функции.	5
				Тема 6. Последовательности.	5
				Тема 14. Библиотеки машинного обучения.	5
				Тема 16. Функциональное программирование.	5
				Тема 1. Базовые средства языка C.	6
				Тема 3. Составной тип данных и динамические структуры.	6
				Тема 14. Сети и протоколы.	6
				Тема 20. Многопоточное программирование в ОС Unix.	6
2	ПК-4	Способен разрабатывать программы и их блоки, проводить их отладку и настройку для решения отдельных задач биотехнических систем	ПК-4.1. Разрабатывает программы и их блоки для решения отдельных задач биотехнических систем ПК-4.2. Проводит отладку и настройку программ для решения отдельных задач биотехнических систем	Тема 2. Операционные системы и виды пользовательского интерфейса.	4
				Тема 5. Язык Паскаль и начала программирования.	4
				Тема 6. Подпрограммы. Процедуры. Функции.	4
				Тема 12. Адреса, указатели и динамическая память.	4
				Тема 17. Возможности процессора и язык ассемблера.	4
				Тема 20. Основные особенности ассемблера NASM.	4
				Тема 2. Путь разработки и использование примеров кода.	5
				Тема 4. Функции. Вызов функции.	5
				Тема 6. Последовательности.	5

				Тема 14. Библиотеки машинного обучения.	5
				Тема 16. Функциональное программирование.	5
				Тема 1. Базовые средства языка C.	6
				Тема 3. Составной тип данных и динамические структуры.	6
				Тема 14. Сети и протоколы.	6
				Тема 20. Многопоточное программирование в ОС Unix.	6

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Индикаторы достижений компетенции (по реализуемой дисциплине)	Перечень планируемых результатов	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1	ОПК-4	ОПК-4.1. Знать принципы работы современных информационных технологий для решения задач профессиональной деятельности ОПК-4.2. Уметь выбирать современные информационные технологии для решения задач профессиональной деятельности ОПК-4.3. Иметь навыки применения современных информационных технологий для решения задач	Знать: принципы работы современных информационных технологий для решения задач профессиональной деятельности. Уметь: выбирать современные информационные технологии для решения задач профессиональной деятельности. Иметь: навыки применения современных информационных технологий для решения задач профессиональной деятельности.	Тема 1 Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8 Тема 9 Тема 10 Тема 11 Тема 12 Тема 13 Тема 14 Тема 15 Тема 16 Тема 17 Тема 18 Тема 19 Тема 20 Тема 21 Тема 22 Тема 23 Тема 24	Вопросы и задания к практической работе, вопросы к зачету, экзамену.

		профессионально й деятельности			
1	ПК-4	ПК-4.1. Разрабатывает программы и их блоки для решения отдельных задач биотехнических систем ПК-4.2. Проводит отладку и настройку программ для решения отдельных задач биотехнических систем	Знать: основы программирования на языках высокого и низкого уровня, для различных операционных систем, с использованием интернет-технологий. Уметь: Разрабатывает программы и их блоки для решения отдельных задач биотехнических систем. Владеть: Проводит отладку и настройку программ для решения отдельных задач биотехнических систем.	Тема 1 Тема 2 Тема 3 Тема 4 Тема 5 Тема 6 Тема 7 Тема 8 Тема 9 Тема 10 Тема 11 Тема 12 Тема 13 Тема 14 Тема 15 Тема16 Тема17 Тема18 Тема 19 Тема 20 Тема 21 Тема 22 Тема 23 Тема 24	Вопросы и задания к практической работе, вопросы к зачету, экзамену.

Фонды оценочных средств по дисциплине «Языки программирования»

Оценочные средства для текущей аттестации (практическая работа):

Вопросы к практическим работам:

1. Зная, что корень n -й степени из x равен x в степени $1/n$, и используя свое понимание использования комплексных чисел в Python, найдите квадратный корень из -17 .
2. Используя инструкции Python, вычислите остаток от деления 17 на 7 и сохраните результат в переменной с именем Z. Затем с помощью одного оператора Python и без скобок умножьте переменную Z на Z+3.
3. Заставьте одну команду Python отображать значение выражения $1.2e+3+34.5$ 20 раз на экране, каждый раз разделяя их точкой с запятой.
4. Какие типы чисел используются в ПИТОНЕ?
5. Как записываются комплексные числа в ПИТОНЕ?
6. Какие основные операции со строками применяются в ПИТОНЕ?
7. Являются ли совместимыми две версии Питона — Python 2 и Python 3?
8. Создайте переменную $a = 10$, чтобы возвести значение в 3 степень и найти остаток от деления на 3 (код программы на ПИТОНЕ).
9. Найдите площадь круга с радиусом 1.23456 (Считаем $\pi = 3.14$).

10. Найдите, истинно ли, что 2 в 10 степени больше 1000 при условии, что 4 в 7 й степени меньше 8 в 5 й.
11. Мы принимаем целое число на ввод с клавиатуры, вывести "цена выше рынка", если число выше 1200 , "цена ниже рыночной", если число меньше 800 и "все ок" в остальных случаях (представить код программы).
12. Распечатайте с 4 по 8 символ фразы "Привет мир!" приведённые к верхнему регистру.
13. Создайте переменную, которая хранит описание этой задачи как строку, разделите слова по пробелам и приведите к нижнему регистру.
14. Напишите код, который проверяет, является ли переменная x строкой, и если да, то если в строке больше одного слова (слово -это последовательность подряд идущих символов без пробелов), то выводит на экран количество слов в этой строке и сами слова в алфавитном порядке.
15. Напишите код, который все элементы массива x с нечетными индексами переставит в обратном порядке.
16. Напишите цикл, который выводит все числа от 0 до 500 , делящиеся на 7 , если в них есть цифра 8 .
17. Создайте список a , состоящий из каких-то элементов. Создайте список b такого же размера, как a , состоящий из каких-то элементов. Выведите нумерованный список пар из элементов списков a и b .
18. Дан массив строк mas . Одной строкой создайте словарь, в котором по ключу строки будет записана пара (кортеж длины 2) (индекс строки в массиве mas , длина строки).
19. Напишите программу `duplicate.py`, которая считывает строку от пользователя, а затем отображает 30 копий этой строки, каждая на отдельной строке.
20. Напишите программу «`pole_tr.py`», которая вычисляет площадь треугольника при условии, что пользователь вводит высоту и длину основания треугольника. Обратите внимание, что высота и длина основания могут быть нецелыми числами.
21. Напишите программу «`odsetki.py`», которая рассчитает баланс счета за N лет, где начальный баланс счета равен SPK , а процентная ставка равна $P\%$ годовых (применяется ежемесячная капитализация).
22. Напишите программу «`even.py`», которая считывает целое число от пользователя и отображает, четное это число или нечетное.
23. Напишите программу «`integer.py`», которая считывает число от пользователя и отображает, является ли оно целым или нецелым.
24. Напишите программу «`prk.py`», которая вычисляет все действительные корни квадратного уравнения вида $ax^2+bx+c=0$, где a , b и c задаются пользователем. Программа должна сначала проверить, является ли введённое уравнение действительно квадратным.
25. Напишите программу «`even2.py`», которая считывает целое число от пользователя и без использования оператора `if` отображает, является ли число четным или нечетным.
26. Напишите программу «`number.py`», которая преобразует строку чисел, введенную пользователем, в текстовую форму:

- a. нецифровые символы следует игнорировать
- b. мы конвертируем цифры, а не числа, поэтому : i. 911 это " девять один один" ii. 1100 это " один один ноль ноль".
27. Вода замерзает при 32 градусах по Фаренгейту и кипит при 212 градусах по Фаренгейту. Напишите программу "grades.py", отображающую таблицу преобразования градусов Цельсия в градусы Фаренгейта в диапазоне от -20 до +40 градусов Цельсия (с шагом 5 градусов). Не забудьте поставить знак плюс/минус рядом с температурой.
28. Напишите программу «ratings.py», которая считывает последовательные оценки пользователя и:
- проверяет, есть ли введенная оценка в списке допустимых оценок в отделе оценок
 - если оценка есть в списке допустимых оценок на отделе оценок, он добавляет ее в список полученных оценок
 - если был нажат только Enter, это означает конец списка полученных оценок
 - отображает среднее арифметическое, рассчитанное для списка полученных оценок.
29. Для чего используется метод rjust() ?
30. Для чего используется метод extend()?
31. Напишите программу «tryg.py», которая будет считывать угол в градусах от пользователя и отображать значение четырех основных тригонометрических функций (sin, cos, tan, ctg), если это возможно для данного угла.
32. Напишите программу «lotto.py», которая будет отображать 6 случайных неповторяющихся чисел от 1 до 49.
33. Напишите программу "words.py", которая будет читать некоторый текст от пользователя, а затем разделять его на предложения (предполагаем, что точка однозначно разделяет предложения) и для каждого предложения будет отображать, сколько в нем слов (предполагаем что пробел разделяет слова в предложении).
34. Какие компоненты должны присутствовать в конструкции определения функции?
35. Напишите функцию, которая будет генерировать случайное предложение, содержащее заданное количество (по умолчанию 5) случайно сгенерированных слов.
36. Напишите программу "liczby_slownie2.py", которая для введенного десятичного числа (в диапазоне 1-1999) будет отображать его значение, записанное словами.
37. Напишите программу „liczby_rzymskie2.py", которая преобразует введенное римское число в его десятичную форму.
38. Как создать пустое множество?

39. Напишите программу `studenci.py`, которая будет использоваться для хранения списка экзаменационных оценок студентов.
40. Напишите программу `«lotto.py»`, которая отображает 6 случайных неповторяющихся чисел от 1 до 49.
41. Измените программу `«lotto»` таким образом, чтобы она отображала случайные числа от наименьшего к наибольшему.
42. Измените программу `"lista studentów"` таким образом, чтобы она дополнительно вычисляла средние оценки всех учеников.
43. Создать новый список на основе нескольких существующих списков.
44. С помощью генератора составьте список, содержащий все числа, кратные 3, от 1 до 33.

Используйте функцию фильтра, чтобы удалить из него все чётные числа, Используя функцию `lambda` и функцию `map`, все элементы результирующего списка возвести в куб.

С помощью функций `reduce` и `len` вычислить среднее арифметическое элементов полученного таким образом списка.

45. Создайте три списка по 5 элементов в каждом:
 - имена - с именами сотрудников,
 - часы - с количеством отработанных часов,
 - ставка - со ставкой в рублях за час работы.используя функции `zip`, `map`, `reduce` и `filter` (и, возможно, генераторы списков), отобразите имена и зарплаты (почасовая заработная плата, умноженная на количество отработанных часов) тех сотрудников, которые заработали больше, чем средняя зарплата.
46. Напишите программу, `"type.py"`, которая будет отображать на экране содержимое файла с именем, заданным пользователем.
47. Напишите программу `lista.py`, которая: - а) Если на диске нет файла `"lista.txt"`, то программа загружает список учеников (имя, фамилия, группа) от пользователя и сохраняет его в файл `"lista.txt"`, - б) Если файл `"lista.txt"` уже есть на диске, то программа загрузит из него список учеников (имя, фамилия, группа), и позволит пользователю добавлять новых учеников в список, и, наконец, сохранить список вернуться к файлу.
48. Напишите программу `«random_file.py»`, которая считывает от пользователя три целых числа: `a`, `b` и `n`, а затем генерирует файл, содержащий `n` строк, каждая из которых содержит случайное целое число от `a` до `b`. В разных линиях могут находиться разные числа, но одно и то же число может встречаться в файле более одного раза.
49. Напишите программу `«random_file2.py»`, которая отличается от программы из упражнения III тем, что указанное число может встречаться во всем файле только один раз. В файле не может быть больше строк, чем длина диапазона от `a` до `b`.
50. Напишите программу, которая найдет все файлы в каталоге, указанном пользователем, и все его подкаталоги, содержащие указанную пользователем строку.

51. Напишите программу, которая находит самый старый и самый длинный файл в заданном каталоге и во всех его подкаталогах.
52. Напишите программу, которая находит все повторяющиеся файлы в заданном каталоге и во всех его подкаталогах, т. е. файлы, имя которых появляется более чем в одном месте одновременно.
53. Напишите программу 'narzedzia.py' для работы со складом инструментов. Программа предназначена для выполнения следующих функций: добавление нового инструмента в список, изменение количества единиц инструмента в списке (в том числе возможность полного удаления инструмента при уменьшении количества штук до нуля), выдача инструмента (запоминание времени и имени работника, сдавшего инструмент), возврат инструмента, список всех инструментов, список доступных инструментов, список отпущенных инструментов.
54. Какая библиотека используется для визуализации информации в ПИТОНЕ?

Критерии и шкала оценивания по оценочному средству текущего контроля (практическая работа)

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	выполнены все задания практической работы, студент чётко и без ошибок ответил на все контрольные вопросы, нет погрешностей в оформлении работы.
хорошо (4)	задания практической работы выполнены с несущественными недочётами или неточностями; студент ответил на все контрольные вопросы с замечаниями, присутствуют некоторые погрешности в оформлении.
удовлетворительно (3)	выполнены все задания практической работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями, допущено небрежность и неточность у оформлении.
неудовлетворительно (2)	Студентом допущены серьёзные ошибки по содержанию работы или задания практической работы выполнены неправильно; студент ответил на контрольные вопросы с ошибками или не ответил на контрольные вопросы.

Оценочные средства для промежуточной аттестации (экзамен):

Вопросы к экзамену:

1. Процессор, память, шина. Принципы работы центрального процессора.
2. Внешние устройства. Иерархия запоминающих устройств.
3. Операционные системы и виды пользовательского интерфейса.
4. Редакторы текстов. Права доступа к файлам.
5. Графическая подсистема в ОС Unix.
6. Позиционные системы счисления. Двоичная логика.
7. Алгоритмы и вычислимость. Алгоритм и его свойства.
8. Машинное представление целых чисел. Числа с плавающей точкой.

9. Понятие цикла `for` PYTHON; оператор `while`. Цикл с постусловием; оператор `break`.
10. Глобальные переменные `for` PYTHON.
11. Концепция структурного программирования.
12. Подпрограммы. Процедуры. Функции.
13. Средства работы с символами в Паскале.
14. Система типов Паскаля.
15. Текстовые файлы. Типизированные файлы.
16. Адреса, указатели и динамическая память.
17. Указатели в Паскале. Динамические переменные.
18. Модули в Паскале. Использование модулей друг из друга.
19. Возможности процессора и язык ассемблера.
20. Правила оформления ассемблерных программ.
21. Основы системы команд `i386`.
22. Организация стека в процессоре `i386`.
23. Основные особенности ассемблера `NASM`.
24. Макросредства и макропроцессор.
25. Многострочные макросы и локальные метки.
26. Макросы с переменным числом параметров.
27. Аппаратная поддержка мультизадачности.
28. Системные вызовы и «программные прерывания».
29. Поддержка модулей в `NASM`. Пример. Объектный код и машинный код.
30. Арифметика с плавающей точкой.
31. Форматы чисел с плавающей точкой. Устройство арифметического сопроцессора.
32. Команды вычисления математических функций. Сравнение и обработка его результатов.
33. Блокноты `Jupyter`. Блокноты `Google Colab`. Текстовые ячейки `Colab`.
34. Инструкции присваивания. Имена переменных. Выражения и инструкции в ПИТОНЕ.
35. Функции. Вызов функции.
36. Математические функции. Композиции. Добавление новых функций. Определение и использование.
37. Параметры и аргументы. Переменные и параметры внутри функций — локальные.
38. Основные типы данных в ПИТОНЕ.
39. Классы, объекты и точечная нотация.
40. Последовательности.
41. Индексирование. Слайсинг.
42. Списки и кортежи. Создание списков и кортежей. Добавление и удаление элементов списка.
43. Словари. Создание словарей. Доступ, добавление и обновление помощью ключей. Удаление элементов из словарей.

44. Множества. Операции над множествами. Непересекающиеся множества. Подмножество. Надмножества. Объединение. Пересечение. Разность.
45. Составные операторы. Структура составных операторов.
46. Операции сравнения. Булевы операции. Оценка объектов. Операторы if.
47. Циклы while. Циклы for. Операторы break и continue.
48. Функции. Объявление функций. Параметры.
49. Область видимости в функциях. Декораторы. Анонимные функции.
50. Установка и импорт библиотеки NumPy.
51. Создание массивов. Индексация и слайсинг.
52. Математические функции NumPy.
53. Библиотека SciPy. Подмодуль scipy.misc. Подмодуль scipy.special. Подмодуль scipy.stats. Дискретные распределения. Непрерывные распределения.
54. Библиотека Pandas. Создание датафреймов из словаря. Создание датафреймов из списка списков. Создание датафрейма из файла. Взаимодействие с данными датафреймов.
55. Библиотека визуализации данных matplotlib. Оформление графиков. Маркировка данных. Построение графиков для множества наборов данных.
56. Библиотека Seaborn. Темы Seaborn. Библиотека Plotly. Библиотека Vokeh. Другие библиотеки визуализации.
57. Популярные библиотеки машинного обучения.
58. Принцип работы машинного обучения.
59. Инструментарий естественного языка (NLTK).
60. Функциональное программирование. Функции функционального программирования. Списковые включения. Базовый синтаксис списковых включений.
61. Замена map и filter. Множественные переменные. Словарные включения. Генераторы. Выражения-генераторы. Функции-генераторы.
62. Объектно-ориентированное программирование.
63. Связывание состояния и функции. Классы и экземпляры. Переменные класса.
64. Сортировка. Списки. Чтение и запись файлов.
65. Системы и сети.
66. Базовые средства языка C.
67. Операторы языка C. Перечислимый тип.
68. Адреса, указатели и операции над ними в языке C.
69. Стандартные функции ввода-вывода в языке C.
70. Макроопределения и макровыводы. Макросы и конструкция do { } while(O).
71. Директивы условной компиляции. Директива #include.
72. Роль и место операционной системы. Управление пользовательскими задачами. Управление внешними устройствами.

73. Файловая система. Права доступа к файлам. Чтение и запись содержимого файлов.
74. Управление объектами файловой системы. Файлы устройств и классификация устройств.
75. Процесс: что это такое. Свойства процесса. Виртуальное адресное пространство. Порождение процесса. Завершение процесса.
76. Ожидание завершения; процессы-зомби. Пример запуска внешней программы. Выполнение процессов и время.
77. Драйвер терминала и дисциплина линии. Сеансы и группы процессов. Управление драйвером терминала. Процессы-демоны.
78. Сети и сетевые соединения. Шлюзы и маршрутизация. Сетевые протоколы. Понятие протокола и модель OSI.
79. Физические канальные протоколы. Физические и канальные протоколы. Протокол IP. Дейтаграммы и потоки. Протоколы прикладного слоя. Доменные имена.
80. Дейтаграммы и потоки. Протоколы прикладного слоя. Доменные имена.
81. Клиент-серверная модель.
82. Параллельное программирование и разделяемые данные.
83. Многопоточное программирование в ОС Unix.
84. Основные принципы работы ОС. Ядро как обработчик запросов. Загрузка и жизненный цикл ОС UNIX.
85. Эмуляция физического компьютера. Структура и основные подсистемы ядра.
86. Управление процессами. Процесс как объект ядра системы. Планирование времени процессора. Обработка сигналов.
87. Управление оперативной памятью.
88. Управление аппаратурой; ввод-вывод. Взаимодействие ОС с аппаратурой. О роли аппаратных прерываний.
89. Буферизация ввода-вывода. Планирование дисковых обменов. Виртуальная файловая система. Файловая система на диске.
90. Компилятор gcc. Средства отладки. Утилита make. Редактор vim.

Критерии и шкала оценивания по оценочному средству промежуточного контроля (экзамен)

Шкала оценивания	Критерий оценивания
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом, дает полное и логически стройное изложение содержания при ответе в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает свои ответы, хорошо владеет умениями самостоятельно обобщать и излагать материал и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути

	излагает его в устной или письменной форме, допуская незначительные неточности в ответах, трактовках и определениях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно чёткие формулировки и непоследовательность в ответах, излагаемых в устной или письменной форме, показывает неусвоение отдельных существенных деталей. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 40% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в определении понятий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы

Форма листа изменений и дополнений, внесенных в ФОС

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)