

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Ректор Белорусского
государственного университета

А.Д.Король

15 июля 2024 г.

Регистрационный № 1882/б.



КОМПЬЮТЕРНАЯ МАТЕМАТИКА

Учебная программа учреждения образования
по учебной дисциплине для специальности:

6-05-0533-07 Математика и компьютерные науки

Профилизация: Искусственный интеллект и математическая экономика

2024 г.

Учебная программа составлена на основе ОСВО 6-05-0533-07-2023, учебного плана БГУ № 6-5.4-55/04 от 15.05 2024.

СОСТАВИТЕЛИ:

Л.Л. Голубева, заведующая кафедрой дифференциальных уравнений и системного анализа механико-математического факультета Белорусского государственного университета, кандидат физико-математических наук, доцент;

О.А. Лаврова, доцент кафедры дифференциальных уравнений и системного анализа механико-математического факультета Белорусского государственного университета, кандидат физико-математических наук;

А.Э. Малевич, доцент кафедры дифференциальных уравнений и системного анализа механико-математического факультета Белорусского государственного университета, кандидат физико-математических наук, доцент;

Н.Л. Щеглова, доцент кафедры дифференциальных уравнений и системного анализа механико-математического факультета Белорусского государственного университета, кандидат физико-математических наук, доцент.

РЕЦЕНЗЕНТЫ:

В.В. Сорокина, математик-программист ООО "Селектио", кандидат технических наук.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой дифференциальных уравнений и системного анализа БГУ
(протокол № 12 от 25.04.2024)

Научно-методическим советом БГУ
(протокол № 9 от 28.06.2024)

Зав. кафедрой дифференциальных уравнений
и системного анализа, доцент



Л. Л. Голубева

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время бурное развитие получило новое, актуальное научное направление – компьютерная математика. Дисциплина «Компьютерная математика» может быть определена как обучающий курс, представляющий совокупность фундаментальных знаний, методов, практик, алгоритмических и компьютерных средств, предназначенных для эффективного решения всех видов задач математического содержания, с высокой степенью визуализации результатов на всех этапах исследований.

Подход к решению задач компьютерными методами заключается в использовании готовых математических сред для исследований – систем компьютерной математики, таких как: *MathCAD*, *MATLAB*, *Mathematica*, *Maple*, *Python* и других, с использованием возможности их интеграции.

Системы компьютерной математики позволяют исследовать проблему, провести анализ и визуализацию данных, построить модели, их тестировать, проверить существование решения, документировать и оформить результаты. При их использовании основное внимание сосредотачивается на существе проблемы и компьютерных моделях для ее разрешения, оставляя в стороне технические вопросы классической математики, детали вычислительных методов и алгоритмических процедур, нюансы языков программирования и команд операционной системы.

Цели и задачи учебной дисциплины

Цель учебной дисциплины – повышение уровня профессиональной компетентности студентов путем обучения навыкам и умениям компьютерного моделирования и проведения исследований для решения проблем классического и современного естествознания.

Образовательная цель: обучение студентов методам и приемам компьютерного моделирования в современных компьютерных математических средах, эффективному исследованию посредством компьютера широкого круга проблем математического содержания.

Развивающая цель: формирование у студентов умений использования существующих и самостоятельной разработки новых технологий компьютерного моделирования.

Задачи учебной дисциплины:

- Исследование и разработка моделей, алгоритмов, методов, программных решений, инструментальных средств по тематике выполняемых научно-исследовательских проектов;
- Разработка и исследование математических, информационных и имитационных моделей в определенной предметной области;
- Составление и подготовка к публикации научных обзоров, рефератов, библиографий, научно-технических отчетов по тематике проводимых исследований.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к модулю «Программирование искусственного интеллекта» компонента учреждения высшего образования.

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

Изучение дисциплины «Компьютерная математика» тесно связано со знаниями, умениями и навыками, полученных при изучении дисциплины «Математический анализ» модуля «Математический анализ 1», дисциплин модуля «Алгебра и геометрия 1», дисциплины «Методы программирования» модуля «Программирование» государственного компонента.

Требования к компетенциям

Освоение учебной дисциплины «Компьютерная математика» должно обеспечить формирование у студентов следующих компетенций:

базовые профессиональные компетенции:

Применять теоретические знания и навыки в самостоятельной исследовательской деятельности.

В результате изучения дисциплины «Компьютерная математика» студент должен для современной компьютерной математической системы *Mathematica*, а также для среды разработки JupyterLab

знать:

- идеологию систем и принципы работы в них; инструментальные средства, элементы управления, интерфейс; структуры данных; особенности построения функций пользователя; возможности визуализации исследований и оформления результатов исследований в виде публикаций;

- особенности языков программирования в соответствующих средах;

уметь:

- применять современный математический аппарат в эффективной интеграции с инструментальными компьютерными математическими средствами;

- создавать и исследовать математические, компьютерные, имитационные модели различных уровней абстракции;

- разрабатывать и анализировать алгоритмы, методы и программные решения по тематике выполняемых исследований;

- квалифицированно применять языки программирования современных систем компьютерной математики;

- проводить анализ результатов исследований, строить информационные модели в средах современных математических пакетов;

- готовить материалы к публикации, в том числе в электронных изданиях, по тематике и результатам проводимых исследований;

- самостоятельно расширять компьютерные математические знания с дальнейшим их использованием при построении и анализе математических и компьютерных моделей широкого круга теоретических и прикладных задач.

иметь навык:

- пользоваться методами и приемами построения моделей объектов, данных, процессов, систем;
- пользоваться методами исследований и решения проблем математического содержания с использованием математических компьютерных приложений.

Структура учебной дисциплины

Дисциплина «Компьютерная математика» изучается в 1-м и 2-м семестрах. Всего на изучение учебной дисциплины «Компьютерная математика» отведено:

1 семестр:

- для очной формы получения высшего образования: 90 часов, в том числе 36 аудиторных часов, из которых лекции – 14 часов + 4 часа ДОТ, лабораторные – 10 часов + 4 часа ДОТ, управляемая самостоятельная работа – 4 часа.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – зачет.

2 семестр:

- для очной формы получения высшего образования: 116 часов, в том числе 68 аудиторных часов, из которых лекции – 30 часов + 4 часа ДОТ, лабораторные – 24 часов + 4 часа ДОТ, управляемая самостоятельная работа – 6 часов.

Трудоемкость учебной дисциплины составляет 3 зачетные единицы.

Форма промежуточной аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Символьный пакет *Mathematica*

Тема 1.1. Сценарий работы в среде символьного математического пакета

Позиционирование дисциплины Компьютерная Математика в системе математических дисциплин. Моделирование как метод научного познания. Особенности компьютерного моделирования. Компьютерные математические среды для моделирования объектов, явлений, процессов.

Числовые и символьные пакеты. Назначение и структура символьных математических пакетов. Интерфейс. Сценарий работы. Сессия (Session) как объект работы Ядра (Kernel). Накопление знаний во время Сессии и хранение знаний между сессиями и сеансами работы с пакетом. Метафора записной книжки (Notebook). Ячейки (Cells) как основные обобщенные объекты Блокнотов. Основной цикл (In[xx]->Out[xx]) работы пакета.

Тема 1.2. Выражение как основная структура данных

Понятие рекурсии. Рекурсивное определение выражения. Атомарные объекты. Функции-конверторы типов атомарных объектов. Уровни выражения, глубина. Позиция подвыражения. Анализ структуры выражения.

Список как наиболее используемое выражение. Встроенные функции для работы со списками.

Тема 1.3. Функциональный стиль программирования

Функции без имени, или анонимные функции. Построение суперпозиции функций без имени. Оператор Apply. Семейство операторов Map. Возможности последовательного применения функций, операторы повторного действия (Nest, Fold, FixedPoint).

Тема 1.4. Образцы

Образец (Patterns) как мета выражение. Основные правила записи. Описание различных множеств выражений посредством использования образцов. Образцы, соответствующие условию. Образцы, содержащие альтернативу. Установка значения выражения по умолчанию. Образцы, допускающие повторение выражений. Встроенное значение части выражения. Построение функции пользователя: образцы при описании ОДЗ функции.

Построение собственной функции дифференцирования с ведением протокола.

Тема 1.5. Основы графики

Графические примитивы. Режимы выполнения графических функций. Базовые функции двумерной и трехмерной графики. Специализированные графические пакеты.

Проектирование функций, осуществляющих преобразование множества точек комплексной плоскости. Координатные сетки на плоскости: прямоугольная декартова, полярная. Роль и задание цвета при построении сетки.

Отображение графического объекта на плоскости, состоящего из точек и линий, при действии на него заданной функцией.

Тема 1.6. Правила преобразований. Глобальные определения

Функции семейства Set. Механизм верхних и нижних значений символа. Программирование, основанное на глобальных правилах преобразований. Способы задания, представления и отображения функций.

Восстановление функции по заданным свойствам. Восстановление Гамма-функции по ее свойствам и значениям в узлах на заданном отрезке.

Тема 1.7 Локальные правила преобразований, подстановки

Определение локальных правил преобразований. Использование образцов. Функции семейства Replace. Подстановки в тождественных преобразованиях алгебраических выражений. Программирование, основанное на локальных правилах преобразований.

Подстановки. Построение псевдо- n -мерного гиперкуба и геометрических фракталов.

Тема 1.8. Визуализация исследований. Динамическая интерактивность

Вычисление выражений внутри функций Dynamic и Manipulate. Интерактивная графика. Построение демонстрационных моделей.

Тема 1.9. Организация вычислительного процесса в *Mathematica*

Роль символа в процессе вычислений. Предопределенные константы. Атрибуты символа. Опции, или режимы выполнения функций. Глобальные определения правил преобразований.

Механизмы локализации символов в определенных границах. Итераторы. Функции Block, Module, With.

Порядок вычисления выражения. Стандартный процесс вычисления выражения. Выражения, вычисляемые нестандартно. Вычисление правил преобразований, глобальных определений. Изменение стандартного порядка вычисления выражения в структурах выбора и повторения.

Конструкции, управляющие ходом вычисления: операторы выбора и операторы повторения.

Раздел 2. Компьютерное моделирование на Python

Тема 2.1. Язык Python. Синтаксические особенности языка. Среда разработки JupyterLab

Основные характеристики языка программирования Python. Применение языка программирования Python. Структура программы на Python. Байт-код. Виртуальная машина Python. Отличие синтаксиса Python от синтаксиса других современных языков программирования. Особенности синтаксиса. Способы выполнения программ, написанных на Python.

Принципы работы с интерактивными документами Jupyter Notebook (ipynb файлы) в среде разработки JupyterLab. Типы ячеек, режимы работы с ячейками. Ядро Python.

Тема 2.2. Объект как фундаментальное понятие в Python. Встроенные типы объектов и операции над ними

Понятие объекта. Атрибуты и методы объекта. Встроенные типы числовых объектов (int, float, complex), встроенный булевый тип (bool), встроенные типы для объектов последовательностей (string, list, tuple), встроенный тип отображение (dict), встроенный тип множество (set). Изменяемые и неизменяемые типы объектов.

Переменная, значение переменной, ссылка на объект. Динамическая типизация переменных. Копирование объектов. Операции и методы для встроенных типов объектов. Выражение спискового включения. Создание строкового объекта на основе подстановок (форматирование строк). Файловый объект.

Тема 2.3. Операторы

Синтаксические правила для составного и составного операторов. Операторы присваивания. Типы присваиваний: базовое, присваивание последовательностей, групповое, дополненное, расширенное. Неявные присваивания. Пространство имен. Оператор выбора. Оператор сопоставления. Операторы цикла. Область видимости переменной. Функция вывода.

Тема 2.4. Функция. Область видимости переменной

Определение функции: операторы def и return, значения по умолчанию для аргументов, произвольное количество аргументов, порядок следования аргументов. Вызов функции: позиционные и ключевые аргументы. Передача значений аргументов через неявное присваивание. Режимы сопоставления объектов-значений и аргументов-переменных при вызове функции. Объект функции. Определение и применение lambda-функций.

Типы областей видимости переменной: локальная область видимости, область видимости объемлющих функций, глобальная область видимости, встроенная область видимости. Правило LEGB для распознавания области видимости имени переменной. Доступ к пространству имен. Операторы объявления global, nonlocal.

Правила оформления строк документации для функций. Аннотация функции.

Анимация движения секущей прямой к заданной линии на плоскости. Пользовательская функция для моделирования границы правильного многоугольника Рело. Анимация качения треугольника Рело.

Тема 2.5. Механизм итераций

Итерируемые объекты. Проход по элементам итерируемого объекта. Объект итератора. Итерационные инструменты. Протокол итераций. Генераторные объекты. Генераторные выражения. Генераторные функции.

Вложенные итерируемые объекты. Задача о выравнивании списка, состоящего из итерируемых объектов.

Тема 2.6. Функциональное программирование

Инструменты функционального программирования. Функции `map`, `filter`, `enumerate`, `zip`. Функция `reduce` из модуля `functools`. Фабричная функция, функция-замыкание, декоратор функции.

Создание текстового файла большого объема с помощью пользовательской генераторной функции.

Тема 2.7. Модуль как важная программная структура

Понятие модуля, пакета, расширения. Модульная структура программ. Импорт модуля: загрузка файла, создание объекта модуля, выполнение байт-кода интерпретатором. Операторы `import` и `from`. Объект модуля. Доступ к переменным импортированного модуля. Перегрузка модуля. Стандартная библиотека модулей Python.

Тема 2.8. Основы объектно-ориентированного программирования

Понятия класса и экземпляра класса. Оператор создания класса. Методы класса. Метод инициализации экземпляра класса. Дерево наследования. Настройка кода. Создание классов на основе встроенных типов. Перегрузка операций и встроенных функций.

Бинарное дерево поиска. Представление. Проектирование классов. Построение бинарного дерева поиска. Перегрузка операции принадлежности `in`. Перегрузка встроенной функции `len`. Обходы бинарного дерева поиска. Центрированный обход. Поиск минимального и максимального значения в дереве.

Проектирование классов. Композиция. Делегирование. Абстрактный класс. Синглтон. Классовые методы. Статические методы. Атрибут-свойство.

Тема 2.9. Пакет `matplotlib` для графического представления данных

Структура пакета `matplotlib`. Создание графических окон и графических областей. Процедурный подход для построения двумерных графиков: оформление графика, текстовые объекты в графической области. Объектно-ориентированный подход для построения изображений. Графические объекты. Методы графических объектов.

Тема 2.10. Расширение `numpy` для численных вычислений

Тип данных массив (`ndarray`) как базовая структура данных расширения `numpy`. Создание массивов. Доступ к элементам и группе элементов массивов. Изменение структуры массивов. Векторизация вычислений с массивами. Операции линейной алгебры. Модуль `linalg`.

Тема 2.11. Программирование с использованием исключений

Изменение потока управления с помощью исключений. Стандартный обработчик исключений. Встроенные исключения. Перехват исключений с помощью оператора `try`. Генерация исключений интерпретатором Python или с помощью оператора `raise`. Оператор `assert` для контроля за ограничениями на переменные, определяемые пользователем. Диспетчер контекстов. Примеры использования исключений.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Очная форма получения высшего образования с применением дистанционных образовательных технологий (ДОТ)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСР	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	Символьный пакет Mathematica. (1 семестр)	18			14		4	
1.1	Сценарий работы в среде символьного математического пакета	2			2			Отчет по лабораторной работе с устной защитой
1.2	Выражение как основная структура данных	2			2			Отчет по лабораторной работе с устной защитой
1.3	Функциональный стиль программирования	2			2			Отчет по лабораторной работе с устной защитой.
1.4	Образцы	2			2		2	Отчет по лабораторной работе с устной защитой, Контрольная работа
1.5	Основы графики	2 (ДОТ)			2 (ДОТ)			Отчет по лабораторной работе с устной защитой
1.6	Правила преобразований. Глобальные определения	2			2			Отчет по лабораторной работе с устной защитой.

1.7	Локальные правила преобразований, подстановки	2					2	Контрольная работа
1.8	Визуализация исследований. Динамическая интерактивность	2 (ДОТ)			2 (ДОТ)			Отчет по лабораторной работе с устной защитой
1.9	Организация вычислительного процесса в <i>Mathematica</i>	2						Устный опрос
2.	Компьютерное моделирование на Python (2 семестр)	34			28		6	
2.1	Язык Python. Синтаксические особенности языка. Среда разработки JupyterLab	4			4			Отчет по лабораторной работе с устной защитой
2.2	Объект как фундаментальное понятие в Python. Встроенные типы объектов и операции над ними	4			2		2	Отчет по лабораторной работе с устной защитой, Контрольная работа
2.3	Операторы	2			2 (ДОТ)			Отчет по лабораторной работе с устной защитой
2.4	Функция. Область видимости переменной	4			4			Отчет по лабораторной работе с устной защитой
2.5	Механизм итераций	2			4			Отчет по лабораторной работе с устной защитой
2.6	Функциональное программирование	2			4		2	Отчет по лабораторной работе с устной защитой, Контрольная работа
2.7	Модуль как важная программная структура	2 (ДОТ)			-			Устный опрос
2.8	Основы объектно-ориентированного программирования	8			6		2	Отчет по лабораторной работе с устной защитой, Контрольная работа
2.9	Пакет matplotlib для графического представления данных	2 (ДОТ)			2 (ДОТ)			Отчет по лабораторной работе с устной защитой

2.10	Расширение numru для численных вычислений	2			-			Устный опрос
2.11	Программирование с использованием исключений	2			-			Устный опрос

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Иванов, О. А. Дискретная математика и программирование в Wolfram Mathematica: для бакалавров: учебное пособие для студентов направлений подготовки 01.03.02 "Прикладная математика и информатика", 01.03.04 "Прикладная математика", а также для студентов других направлений, изучающих дисциплину "Дискретная математика" / О. А. Иванов, Г. М. Фридман. - Санкт-Петербург [и др.]: Питер, 2020. - 349 с. - URL: <https://ibooks.ru/bookshelf/359233>.
2. Криволапов, С. Я. Математика на Python : учебник для направлений бакалавриата "Экономика" и "Бизнес-информатика" / С. Я. Криволапов, М. Б. Хрипунова ; Финансовый ун-т при Правительстве РФ. - Москва : Кнорус, 2021. - 455 с.
3. Федоров, Д. Ю. Программирование на Python : учебное пособие для вузов, для студ., обуч. по инженерно-техническим направлениям / Д. Ю. Федоров. - 6-е изд., перераб. и доп. - Москва : Юрайт, 2025. - 187 с.

Дополнительная литература

1. Голубева, Л. Л. Компьютерная математика. Символьный пакет Mathematica: курс лекций / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. Минск: БГУ, 2005. 103 с.
2. Вавилов Н.А. Mathematica для нематематика: Учеб. пособие для ВУЗов. / Н.А. Вавилов, В.Г. Халин, А.В. Юрков. М: Московский центр непрерывного математического образования, 2021. 483 с.
3. Седов, Е. С. Основы работы в системе компьютерной алгебры Mathematica : учебное пособие / Е. С. Седов. — 2-е изд. — Москва : ИНТУИТ, 2016. — 401 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100339>
4. Воробьев, Е. М. Введение в систему символьных, графических и численных вычислений "Математика - 5": учеб. пособие для студ. вузов / Е.М. Воробьев. – Москва : Диалог-Мифи, 2005 – 365с. : ил. ; 20х14 см. – ISBN 5-86404-199-8.
5. Lynch, S. Dynamical Systems with Application using Mathematica / S. Lynch. Birkhäuser Boston, 2007. 499 p.
6. Wickham-Jones, T. Mathematica Graphics. Techniques & Applications / T. Wickham-Jones. Springer Verlag New York Berlin Heidelberg, 1994, XIV, 721 p.
7. Sal Mangano. Mathematica Cookbook / Sal Mangano. O'Reilly Media, 2010. 832 p.
8. Wolfram, S. Mathematica book: 5 ed. / S. Wolfram. Wolfram, 2003. 1301 p.

9. Лутц, М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. / М. Лутц. СПб.: ООО “Диалектика”, 2019. 832 с.
10. Лутц, М. Изучаем Python, том 2, 5-е изд.: Пер. с англ. / М. Лутц. СПб.: ООО “Диалектика”, 2020. 720 с.
11. Свейгарт, Эл. Python. Чистый код для продолжающих / Э. Свейгарт; [пер. с англ. Е. Матвеев]. Санкт-Петербург; Москва; Минск: Питер, 2022. – 383 с.
12. Копец, Дэвид. Классические задачи Computer Science на языке Python = Classic Computer Science Problems in Python / Дэвид Копец ; [пер. с англ. Е. Полонской]. - Санкт-Петербург [и др.] : Питер, 2021. - 252 с.
13. Марк Саммерфилд, Программирование на Python 3. Подробное руководство: Символ-Плюс, 2009, 608 с.
14. Уэс Маккинни, Python и анализ данных: ДМК Пресс, 2015, 482 с.
15. Лутц, М. Программирование на Python, том 1, 4-е изд.: Пер. с англ. / М. Лутц. СПб.: Символ-Плюс, 2011. 992 с.
16. Лутц, М. Программирование на Python, том 2, 4-е изд.: Пер. с англ. / М. Лутц. СПб.: Символ-Плюс, 2011. 992 с.
17. Голубева, Л. Л. Компьютерная математика. Символьный пакет Mathematica [Электронный ресурс] : электронный учебно-методический комплекс для специальности: 1-31 03 01 «Математика (по направлениям)», направление специальности: 1-31 03 01-02 «Математика (научно-педагогическая деятельность)» / Л. Л. Голубева [и др.] ; БГУ, Механико-математический фак., Каф. дифференциальных уравнений и системного анализа. – Минск: БГУ, 2021. Ссылка на ресурс: <https://elib.bsu.by/handle/123456789/276777>.
18. Голубева, Л. Л. Компьютерная математика. Символьный пакет МАТНЕМАТИСА : лабораторный практикум для студ. мех.-мат. фак. : в 2 ч. Ч. 1 / Л. Л. Голубева, А. Э. Малевич, Н. Л. Щеглова. - Минск : БГУ, 2012. - 235 с. - URL: <http://elib.bsu.by/handle/123456789/95686>.
19. Чичурин, А. В. Применение системы МАТНЕМАТИСА при решении дифференциальных уравнений и в задачах математического моделирования : курс лекций для студ. спец. 1-31 03 01 "Математика (по направлениях)" : в 3 ч. / А. В. Чичурин, Е. Н. Швычкина ; БГУ, Механико-математ. факультет. – Минск : БГУ, 2016 – 2017.

Рекомендуемое учебно-лабораторное оборудование

Для проведения занятий требуются следующее программное обеспечение: операционная система MS Windows, MS Office, Mathematica, Python, JupyterLab.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Объектом диагностики компетенций студентов являются знания, умения, полученные ими в результате изучения учебной дисциплины. Выявление

учебных достижений студентов осуществляется с помощью мероприятий текущего контроля и промежуточной аттестации.

Для диагностики компетенций могут использоваться следующие средства текущего контроля: отчет по лабораторной работе с устной защитой, контрольная работа и устный опрос.

При защите лабораторных работ оценивается полнота ответа, аргументация выбранных решений, последовательность и оригинальность изложения материала, оригинальность кода, корректность оформления, самостоятельность выполнения заданий.

Формой промежуточной аттестации по дисциплине «Компьютерная математика» учебным планом предусмотрен зачет в 1-м семестре и экзамен во 2-м семестре.

Зачет по дисциплине «Компьютерная математика» в 1-м семестре проходит в форме контрольного опроса в устной или письменной форме, выполнения заданий на компьютере. Если студент успешно защитил все лабораторные работы и получил отметку 4 и выше по контрольным работам, то допускается определение результатов промежуточной аттестации по дисциплине на основании результатов текущего контроля знаний без проведения дополнительного опроса на зачете. При этом явка обучающегося является обязательной.

Экзамен по дисциплине «Компьютерная математика» во 2-м семестре проходит в форме письменного контрольного опроса и далее в устной форме.

Для формирования итоговой отметки по учебной дисциплине используется модульно-рейтинговая система оценки знаний студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущей и промежуточной аттестации студентов по учебной дисциплине.

Формирование итоговой отметки в ходе проведения контрольных мероприятий текущей аттестации (примерные весовые коэффициенты, определяющие вклад текущей аттестации в отметку при прохождении промежуточной аттестации):

- отчеты по лабораторным работам с устной защитой, устные опросы – 60%;
- контрольные работы – 40%.

Итоговая отметка по дисциплине рассчитывается на основе итоговой отметки текущей аттестации (рейтинговой системы оценки знаний) _40_% и экзаменационной отметки 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

Тема 1.4. Образцы (2 часа)

Задание. Построить функцию пользователя $\text{Dif}[\text{expr}, x]$, вычисляющую производную функции одной переменной.

- 1) Изучить встроенные функции дифференцирования, особенности использования каждой.
- 2) Проектировать операторы дифференциального исчисления функций одной переменной.
- 3) Создать глобальные определения для дифференцирования функции одной переменной: правила дифференцирования, производные элементарных функций, производная сложной функции.
- 4) Тестировать созданные функции пользователя.
- 5) Проектировать и построить функцию, вычисляющую производную функции одной переменной и ведущую в процессе вычисления протокол выполнения операций дифференцирования.

Целью данного задания является выработка у студента навыков функционального стиля программирования на языке Wolfram Language.

Форма контроля – отчет по лабораторной работе с устной защитой.
Контрольная работа.

Тема 1.7. Локальные правила преобразований, подстановки (2 часа)

Задание. Написать оператор, генерирующий из отрезка заданный элементарный фрактальный фрагмент. Используйте технику локальных правил преобразований и операторы повторного действия.

Выполнить следующие задачи:

1. Написать функцию фрактального преобразования $\text{fracLine}[\{A:\{_,_\}, B:\{_,_\}\}]$, которая генерирует из произвольного отрезка AB на плоскости указанный элементарный фрактальный фрагмент, согласно варианту;
2. Написать функцию $\text{fracShow}[\{A:\{_,_\}, B:\{_,_\}\}, n_Integer?Positive]$, которая из графического объекта $\text{Line}@\{A,B\}$ для отрезка AB генерирует графический объект для элементарного фрактального объекта $\{\text{Line}@\{P_1,P_2\}, \text{Line}@\{P_2,P_3\}, \text{Line}@\{P_3,P_4\}, \dots, \text{Line}@\{P_{m-1},P_m\}\}$, выполняя n итераций фрактальных преобразований fracLine к каждому звену фрактального фрагмента;
3. Генерировать и отобразить геометрический фрактал, выбирая в качестве основы или отрезок, или равносторонний треугольник, или квадрат, ориентация – внутрь или наружу фигуры, согласно варианту.

Целью данного задания является выработка у студента навыков объектно-ориентированного проектирования, функционального стиля программирования на языке Wolfram Language.

Форма контроля – контрольная работа.

Тема 2.2. Объект как фундаментальное понятие в Python. Встроенные типы объектов и операции над ними (2 часа)

Задание 1. Выполните упражнения 1 – 11, представленные в [Лутц, М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. / М. Лутц. СПб.: ООО “Диалектика”, 2019. 832 с.; с. 340 – 342]. Содержимое структурируйте. Выполнение сопровождайте краткими комментариями, располагая их или в ячейках кода, или в ячейках Markdown.

Задание 2. Проверьте приобретенные знания при изучении, ответив на контрольные вопросы, которые находятся в [Лутц, М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. / М. Лутц. СПб.: ООО “Диалектика”, 2019. 832 с.; часть II], в конце каждой из глав 4 – 9.

Задания выполняются на основе методических указаний к лабораторной работе.

Целью УСР является изучение студентом встроенных типов объектов в Python, способов их создания и операций над ними; укрепление навыков структурирования интерактивных электронных блокнотов в JupyterLab.

Форма контроля – контрольная работа.

Тема 2.6. Функциональное программирование (2 часа)

Задание 1. Напишите пользовательскую функцию `func1` с произвольным количеством ключевых аргументов одного типа (числовые типы, строки, кортежи, списки), которая возвращает результат применения операции сложения ко всем аргументам функции.

Задание 2. Напишите генераторную функцию `func2` для генерации членов геометрической прогрессии $b_1 = 3$, $b_{n+1} = 5 b_n$, $n=1, 2, \dots$. На основе генераторной функции создайте кортеж из 10 первых членов геометрической прогрессии. Вычислите 10-ый член геометрической прогрессии с помощью функции `reduce` из модуля `functools`.

Задание 3. Напишите пользовательскую функцию `func3` с произвольным количеством позиционных и ключевых аргументов. Функция `func3` должна возвращать кортеж, состоящий из трех элементов. Первый элемент кортежа содержит число, соответствующее количеству вызовов пользовательской функции `func3`. Второй элемент кортежа содержит количество позиционных аргументов при вызове функции. Третий элемент кортежа содержит список ключей для ключевых аргументов при вызове функции `func3`. Напишите две реализации счетчика вызовов пользовательской функции `func3`: с использованием глобальной переменной и с использованием синтаксиса функций-замыканий.

Задания выполняются на основе методических указаний к управляемой самостоятельной работе.

Целью УСР является выработка у студента навыков проектирования и написания функций, функционального стиля программирования на Python.

Форма контроля – контрольная работа.

Тема 2.8. Основы объектно-ориентированного программирования (2 часа)

Задание 1. Определите класс `Complex`, экземплярами которого являются комплексные числа. Определение класса должно содержать: метод

инициализации экземпляра класса по действительной и мнимой части комплексного числа со значениями по умолчанию для каждого аргумента; метод, который для экземпляра класса `Complex` возвращает комплексно-сопряженное число в виде нового экземпляра класса `Complex`; метод `abs`, который возвращает модуль комплексного числа. Создайте один экземпляр класса `Complex` и осуществите вызов каждого метода.

Задание 2. Для работы со структурой данных стек определите класс `Stack` на основе встроенного класса `list` с одним методом вставки элемента в стек. Создайте экземпляр класса `Stack` и осуществите вставку элементов, генерируемых функцией `range(100)`.

Задание 3. Изобразите бинарное дерево поиска, соответствующее последовательной вставке в дерево элементов списка `[1,2,3,1,2,3,1,2,3]`. Напишите определение классов `BinaryTree`, `BinaryNode`, `EmptyNode` только с использованием методов инициализации. Поясните назначение каждого атрибута, определенного внутри методов инициализации. Поясните утверждение, что классы `BinaryTree`, `BinaryNode`, `EmptyNode` связаны на основе композиции. Можно ли в бинарное дерево поиска, созданное на основе элементов списка `[1,2,3,1,2,3,1,2,3]` с помощью классов `BinaryTree`, `BinaryNode`, `EmptyNode`, добавить элементы со значениями `True` или `False`? Ответ поясните.

Задание 4. Определите класс, который содержит атрибут класса, метод перегрузки операции, статический метод и классовый метод. Осуществите доступ к атрибуту класса и вызов каждого метода.

Задания выполняются на основе методических указаний к управляемой самостоятельной работе.

Целью УСР является выработка у студента навыков объектно-ориентированного проектирования, объектно-ориентированного стиля программирования на Python.

Форма контроля – контрольная работа.

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используется **эвристический подход**, который предполагает демонстрацию многообразия решений большинства профессиональных задач и жизненных проблем.

При организации образовательного процесса используется **практико-ориентированный подход**, который предполагает освоение содержания через решения практических задач.

При организации образовательного процесса **используются методы и приемы развития критического мышления**, которые представляют собой систему, формирующую навыки работы с информацией в процессе чтения и письма; понимании информации как отправного, а не конечного пункта критического мышления.

На лекциях и лабораторных занятиях используются следующие методы обучения: проблемного изложения, поисковый, репродуктивный,

исследовательский. При проведении занятий также планируется использовать наглядные методы, такие как иллюстрация, демонстрация, визуализация.

Методические рекомендации по организации самостоятельной работы обучающихся

Для организации самостоятельной работы студентов по учебной дисциплине рекомендовано разместить на образовательном портале или сайте кафедры учебно-методические материалы: курсы лекций и лабораторные практикумы, методические указания к лабораторным занятиям, вопросы для подготовки к зачету и экзамену, перечень рекомендуемой литературы, информационные ресурсы.

Самостоятельная работа студента включает в себя работу с учебной литературой по заданным разделам дисциплины, поиск в Интернете новейшей учебной и научной информации в указанных областях знаний и знакомство с ней, а также выполнение задач, поставленных на занятиях.

Примерный перечень вопросов к зачету

1. Составные части Mathematica, назначение каждой. Приобретение знаний Mathematica во время сеанса работы Session.
2. Особенности интерфейса пользователя Mathematica. Структура системы справки. Сообщения и предупреждения.
3. Рабочий документ Mathematica: структура, форматирование, редактирование. Служебные и горячие клавиши. Особенности ввода: типы ячеек, математические выражения, специальные символы.
4. Главный цикл the Main Loop обработки выражения In[] – Out[].
5. Рекуррентное определение выражения. Типы атомарных объектов. Извлечение части атомарного выражения.
6. Полная, или каноническая форма записи выражения. Функции, отображающие структуру выражения.
7. Структура выражения. Определения понятий уровня выражения, глубины выражения, позиции подвыражения.
8. Функции Level, Depth, Position. Спецификатор уровней выражения levelspec, его использование.
9. Функции для извлечения частей выражения, не являющегося атомарным: Extract, Take, Part, Select, Cases, DeleteCases.
10. Списки как основное используемое выражение. Создание списка, изменение его структуры, порядка следования элементов, добавление и удаление элементов списка.
11. Функция без имени: определение, построение, использование. Формы записи безымянной функции. Понятие выражения-однотрочника, правила его написания.
12. Образец Pattern: определение, назначение, способы именования. Функция для сравнения выражения с образцом.

13. Виды образцов: с условием, содержащие альтернативу, повторяющиеся, с заданными по умолчанию значениями, со встроенными по умолчанию значениями.
14. Использование образцов в аргументах встроенных функций Mathematica и в функциях пользователя.
15. Базовые функции двумерной и трехмерной графики.
16. Двумерные графические примитивы. Директивы. Опции.
17. Основные трехмерные графические примитивы. Директивы. Опции.
18. Правила преобразований: определение, классификация, сфера видимости и момент времени вычисления. Использование правил.
19. Пополнение списков значений символа при опеределении глобальных правил преобразований. Место глобального правила в списке.
20. Локальные правила преобразований, или подстановки.
21. Функции для выполнения локальных правил преобразований. Программирование в стиле локальных правил.
22. Функции семейства Map: назначение, аргументы, общие принципы работы. Особенности использования каждой функции. Назначение и использование функции Apply.
23. Функции Map, MapAt, MapAll, MapThread, MapIndexed: назначение, входы и выходы. Примеры.
24. Итерационные процессы. Операторы семейств Nest и Fold. Их назначение, общее и различие между этими группами.
25. Структуры выбора и структуры повторения, их реализация в Mathematica.
26. Локализация сферы видимости переменных. Механизмы локализации символов в структурах Block, Module, With, Function.
27. Стандартный порядок вычисления выражения ядром. Изменение стандартного порядка вычисления выражения при наличии атрибутов семейства xxHoldxxx.
28. Атрибуты: назначение, установка, момент вычисления на примере установленных свойств символа Plus. Особенности вычисления символа, который наделен атрибутом семейства Hold.
29. Динамическая визуализация выражений. Функции Manipulate и Dynamic.
30. Динамическая визуализация графических объектов. Элементы управления, их использование.

Примерный перечень вопросов к экзамену

1. Свойства и особенности языка программирования Python. Стандартная библиотека Python. Концептуальная структура программы. Файловая структура программы.
2. Python как интерпретируемый язык программирования. Трансляция в байт-код. Виртуальная машина Python. Расширения Python.

3. Объект. Встроенные типы объектов. Литералы и конструкторы для создания объектов встроенных типов. Изменяемые и неизменяемые типы. Булево значение объекта.
4. Последовательность. Встроенные типы объектов последовательностей. Операции над последовательностями: индексация, конкатенация, повторение, длина, проверка принадлежности.
5. Словарь. Создание словаря. Операции над словарем. Методы словаря.
6. Списковое включение и его аналоги. Списковое включение с конструкцией выбора. Списковые включения с несколькими уровнями вложенности. Примеры.
7. Файловый объект. Создание файлового объекта. Методы файловых объектов. Итерирование файлового объекта.
8. Строки документации и комментарии. Правила оформления строк документации. Атрибут `__doc__`. Инструмент `PyDoc`.
9. Синтаксический шаблон составного оператора. Оператор выбора `if`. Операторы цикла `for` и `while`.
10. Переменная, ее создание, использование и удаление. Оператор присваивания и его формы. Неявные присваивания. Пространства имен.
11. Пользовательская функция. Оператор определения функции `def`. Определение функции со значениями по умолчанию для аргументов. Определение функции с произвольным количеством аргументов. Объект функции. Аннотация типов при определении функции.
12. Позиционные и ключевые аргументы при вызове функции. Позиционный и ключевой режимы сопоставления объектов-значений и аргументов-переменных при вызове функции.
13. Определение и применение `lambda`-функций. Примеры.
14. Переменная. Типы областей видимости переменной. Правило лексических областей при поиске переменной. Операторы объявления `global` и `nonlocal`.
15. Механизм итераций. Итерируемый объект и его элементы. Объект итератора.
16. Механизм итераций. Ручное и автоматическое выполнение итераций. Итерационные инструменты. Протокол итераций.
17. Генераторный объект. Генераторное выражение. Генераторная функция. Операторы `yield` и `yield from`. Преимущества генераторных объектов перед коллекциями.
18. Функции `map` и `filter` для организации циклов на итерируемых объектах. Примеры.
19. Спецификации функций `enumerate` и `zip`. Использование функций `enumerate` и `zip` в циклах `for`. Использование функции `zip` для работы со словарями.
20. Функция-замыкание. Фабричная функция. Пример определения функции-замыкания со счетчиком вызовов функции.

21. Модуль. Пакет. Операторы импортирования модуля `import` и `from`. Загрузка и перезагрузка модуля.
22. Модуль. Глобальная область видимости переменной. Объект модуля. Пространство имен объекта модуля. Атрибут `__name__` объекта модуля.
23. Оператор создания класса `class`. Объект класса. Экземпляр класса. Атрибуты и методы класса. Объект связанного метода. Концепция инкапсуляции. Метод инициализации экземпляра класса.
24. Дерево наследования. Суперклассы. Подклассы. Множественное наследование. Настройка кода за счет замены метода. Настройка кода за счет расширения кода метода.
25. Перегрузка операций и встроенных функций. Методы строкового представления. Метод вызова. Методы операции сложения.
26. Перегрузка операций и встроенных функций. Методы операций сравнения. Методы булевого значения. Методы операций индексации.
27. Исключение. Операторы для работы с исключениями. Стандартный обработчик исключений. Встроенные исключения. Пользовательские исключения. Диспетчер контекста.
28. Исключение. Оператор `try` для перехвата и обработки исключений. Операторы `raise` и `assert` для генерации исключений.
29. Расширение `numpy`. Тип данных `ndarray`. Атрибуты массива. Итерирование массивов. Создание массивов. Индексация массивов.
30. Расширение `numpy`. Методы для изменения структуры массива. Атрибут `shape`. Представление и копия массива. Атрибут `base`.
31. Расширение `numpy`. Векторизация вычислений. Представление вектора и матрицы с помощью массива. Матричное произведение. Скалярное произведение. Транспонирование матрицы.
32. Пакет `matplotlib`. Базовые функции двумерной графики. Функции для создания и оформления текстовых объектов в графической области.
33. Пакет `matplotlib`. Графические объекты для представления изображения. Иерархическая структура изображения. Объектно-ориентированный подход для построения изображения.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы УВО по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Дифференциальные уравнения	Кафедра дифференциальных уравнений и системного анализа	Изменений не требуется.	Протокол № 12 от 25.04.2024

Зав. кафедрой дифференциальных уравнений
и системного анализа



Л. Л. Голубева

25.04.2024

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
____ (протокол № ____ от ____ 202_ г.)
(название кафедры)

Заведующий кафедрой

(ученая степень, ученое звание)

(И.О.Фамилия)

УТВЕРЖДАЮ
Декан факультета

(ученая степень, ученое звание)

(И.О.Фамилия)