Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Байханов Исмаил Бахтиярович
Должность: Ректор
Дата подписания: 11.02.2025 16:58:56
Уникальный программный ключ:
442c337cd125e1d014f6216988d813e502637764

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ):

«Программирование»

1. Цель освоения дисциплины (модуля):

Цель дисциплины – формирование у обучающихся системы понятий, знаний, умений и навыков в области современного программирования, включающего в себя методы проектирования, анализа и создания программных продуктов, основанные на использовании процедурной и объектно-ориентированной методологии.

Задачи дисциплины:

1. Формирование представлений о современных языках программирования.
2. Формирование умения выполнить структурную декомпозицию предметной области и реализовать разработанные алгоритмы на языке программирования высокого уровня Python для обработки исходных данных различных типов.
3. Формирование умения выполнять отладку программы.
4. Обучение теоретическим основам объектно-ориентированного анализа, проектирования и программирования.
5. Формирование у обучающихся умения выполнять объектную декомпозицию предметной области и разрабатывать иерархию классов, реализуя механизмы наследования и полиморфизма;

Формирование у обучающихся представлений о современных технологиях разработки программных средств информационных технологий.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Программирование» (Б1.О.07.05) относится к предметно-методическому модулю по профилю «Информатика» обязательной части Блока 1 основной профессиональной образовательной программы по направлению подготовки 44.03.01 Педагогическое образование, профиль «Информатика (VR / AR технологии и программирование)». Дисциплина (модуль) изучается на 1-2 курсе, в 1,2,3 семестрах.

Требования к предварительной подготовке обучающегося:

Для освоения дисциплины, обучающиеся используют знания, умения, сформированные в ходе изучения дисциплин: «Архитектура ПК и периферийные устройства», «Основы алгоритмизации и программирования», «Операционные системы».

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

Дисциплина «Программирование» формирует знания, умения и компетенции, необходимые для освоения следующих дисциплин: для проведения следующих практик: «Педагогические программные средства».

3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины направлен на формирование и развитие компетенций:

Код и наименование компетенции	Код и наименование индикатора достижения компетенций, которые формирует дисциплина (модуль)	Планируемые результаты обучения
ПК-1. Способен осваивать и использовать теоретические знания и практические умения и навыки в предметной области при решении	ПК-1.1. Знает структуру, состав и дидактические единицы предметной области (преподаваемого предмета).	Знает: • принципы, базовые концепции технологий программирования; • категории типов данных; • теоретические основы объектно-ориентированного анализа, проектирования программирования; Умеет:

профессиональных задач		• реализовывать и базовые алгоритмические конструкции при составлении программы; • создавать программы для обработки данных различных типов; • выполнять объектную декомпозицию предметной области и разрабатывать иерархию классов, реализуя механизмы наследования и полиморфизма; Владеет: • типовыми алгоритмами обработки данных элементарных и структурированных типов.
	ПК-1.2. Умеет осуществлять отбор учебного содержания для его реализации в различных формах обучения в соответствии с требованиями ФГОС ОО.	Знает: • основные этапы и принципы создания программного продукта; • типы ошибок и способы отладки программы; • абстракции основных структур данных, методы их обработки и способы реализации в объектно-ориентированных программных средах; Умеет: • создавать программы для обработки данных различных типов; • находить и исправлять синтаксические и семантические ошибки в процессе отладки программы; • создавать объекты разработанных классов (в том числе массивы и списки из них) и применять к ним методы; • адаптировать приложения к меняющимся условиям функционирования; Владеет: • технологиями программирования в объектно-ориентированных программных и операционных средах;
	ПК-1.3. Демонстрирует умение разрабатывать различные формы учебных занятий, применять методы, приемы и технологии обучения, в том числе информационные.	Знает: • принципы, базовые концепции технологий программирования; • методы объектно-ориентированного проектирования и программирования; Умеет: • представлять и защищать разработанный программный проект; • разрабатывать спецификацию и документацию разработанного приложения; Владеет: • логическим, алгоритмическим операциональным, и объектным стилями мышления; • языками объектно-ориентированного программирования.

4. Общая трудоемкость дисциплины (модуля) составляет 10 зачетных единиц (360 часов)

5. Основные разделы дисциплины (модуля):

1 семестр

Раздел 1. Структурное программирование. Типизация данных. Синтаксис, основные операторы языка программирования Python.

Введение в программирование. Кроссплатформенность Python: исходный код, байт-код, исполнение программы на Python, PVM. Интегрированные среды разработки на Python, онлайн компиляторы.

Язык программирования как формальный язык. Алфавит языка Python. Идентификаторы в Python. Системные имена.

Структура Python-программы: модули, инструкции, выражения, объекты. Правила записи программы. Отступы как элемент синтаксиса. Переменные, характеристики переменных. Динамическая строгая типизация в Python. Оператор присваивания. Организация ввода/вывода данных.

Базовые типы данных и операции над ними. Категории типов данных в Python: числа, последовательности, множества, словари. Изменяемые и неизменяемые типы данных. Операции на данными. Приоритет операций.

Числовой тип данных в Python. Числовые литералы в Python. Встроенные модули и функции. Инструкция import.

Базовые алгоритмические конструкции и их реализация в языке Python. Условный оператор. Трехместное выражение if/else. Операторы циклов. Цикл while и цикл for. Необязательная часть else. Функция range(). Инструкции pass, break, continue.

Глава 2. Функции и модули в языке Python.

Функции как основные программные структуры в языке Python, обеспечивающие многократное использование программного кода и уменьшающие его избыточность.

Создание пользовательской функции. Оператор def как исполняемый оператор. Оператор return. Вызов функции. Формальные и фактические параметры. Глобальные и локальные переменные. Инструкции global и nonlocal.

Рекурсивные функции. Прямая и косвенная рекурсия. Глубина рекурсии.

Функции высшего порядка. Модуль functools. Декораторы функций @cache и @lru_cache.

Лямбда-функции и анонимные функции в языке Python.

Создание пользовательских модулей.

2 семестр

Глава 3. Обработка строк, списков, множеств и словарей в Python программе. Работа с файлами.

Последовательности как категория типов данных в языке Python. Операции над последовательностями. Обращение по индексу, отрицательные индексы. Срезы.

Класс str. Строковые литералы. Экранированные последовательности. Операции над строками, функции и методы для работы со строками в Python программе. Типовые алгоритмы обработки строк.

Класс list. Списки. Литералы списков. Операции над списками, функции и методы для работы со списками в Python-программе. Создание копии списка и присваивание списков. Вложенные списки. Генераторы списков в языке Python. Кортежи. Типовые алгоритмы обработки списков. Сортировка.

Функции высшего порядка при обработке последовательностей (map(), filter()).

Класс set. Реализация множественного типа в языке Python. Операции над множествами, функции и методы для работы с множествами в Python программе.

Класс dict. Отображения (словари). Создание словаря. Обращение к элементу по ключу. Операции, функции и методы для обработки словарей в Python-программе. Типовые алгоритмы.

Обработка файлов в Python-программе. Текстовые и бинарные файлы. Режимы доступа к файлу. Функция open(). Операции с файлами. Чтение/запись данных. Позиционирование файлового указателя. Модуль os для работы с каталогами и файлами.

3 семестр

Глава 4. Объектно-ориентированный анализ предметной области и проектирование приложений.

Класс. Атрибуты. Поля. Методы, формальный параметр self. Конструктор. Функция __init__(). Объект как экземпляр класса. Создание класса. Оператор class. Создание объекта. Вызов конструктора. Деструктор. Обращение к атрибутам. Перегрузка операторов.

Списки объектов.

Принципы ООП. Наследование. Инкапсуляция. Полиморфизм. Наследование в Python. Суперкласс (родительский) и подкласс (дочерний класс). Классы-предки и потомки. Класс object. Создание дочернего класса. Конструктор дочернего класса. Оператор super(). Переопределение методов родительского класса в дочернем классе. Вызов унаследованных методов. Полиморфизм. Области видимости. Множественное наследование. Проверка принадлежности объекта классу: функция isinstance() и атрибут __class__.

Ассоциация как тип отношения между классами. Объект как поле класса. Виды ассоциации: композиция и агрегация классов.

Исключения и их обработка. Понятие исключения. Виды исключений. Обработка исключений.

Глава 5. Реализация принципов объектно-ориентированного программирования в процессе разработки приложений.

Решение задач на разработку иерархии классов, реализующей отношения различных типов: наследование, композицию, агрегацию.

Разработка и защита итогового проекта.

6. Формы текущего контроля успеваемости и промежуточной аттестации – в 1,3 семестрах экзамен, во 2 семестре зачет.

7. Автор: Исраилов Р.Ю., кан. техн. наук.

Программа одобрена на заседании кафедры ИТ и МПИ,

протокол №8 от 25.04.2024г.

И.о. заведующего кафедрой _____ Исраилов Р.Ю. кан. техн. наук