

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Алтайский государственный технический университет им. И.И. Ползунова»

СОГЛАСОВАНО

Декан ФИТ

А.С. Авдеев

Рабочая программа дисциплины

Код и наименование дисциплины: **Б1.В.13 «Программирование кроссплатформенных систем»**

Код и наименование направления подготовки (специальности): **12.03.01
Приборостроение**

Направленность (профиль, специализация): **Искусственный интеллект в приборостроении**

Статус дисциплины: **часть, формируемая участниками образовательных отношений**

Форма обучения: **очная**

Статус	Должность	И.О. Фамилия
Разработал	заведующий кафедрой	А.Г. Зрюмова
Согласовал	Зав. кафедрой «ИТ»	А.Г. Зрюмова
	руководитель направленности (профиля) программы	А.Г. Зрюмова

г. Барнаул

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Содержание компетенции	Индикатор	Содержание индикатора
ПК-12	Способен разрабатывать программы и их блоки, проводить их отладку и настройку для построения интеллектуальных систем и приборов	ПК-12.1	Разрабатывает программы и их блоки для построения интеллектуальных систем и приборов
		ПК-12.2	Проводит отладку и настройку программ и программного обеспечения для построения интеллектуальных систем и приборов

2. Место дисциплины в структуре образовательной программы

Дисциплины (практики), предшествующие изучению дисциплины, результаты освоения которых необходимы для освоения данной дисциплины.	Дизайн в приборостроении, Информатика, Информационные технологии, Функциональное и логическое программирование
Дисциплины (практики), для которых результаты освоения данной дисциплины будут необходимы, как входные знания, умения и владения для их изучения.	Обработка и преобразование измерительных сигналов, Подготовка к процедуре защиты и защита выпускной квалификационной работы, Системы технического зрения

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающегося с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающегося

Общий объем дисциплины в з.е. /час: 3 / 108

Форма промежуточной аттестации: Экзамен

Форма обучения	Виды занятий, их трудоемкость (час.)				Объем контактной работы обучающегося с преподавателем (час)
	Лекции	Лабораторные работы	Практические занятия	Самостоятельная работа	
очная	16	16	0	76	38

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

Форма обучения: очная

Семестр: 6

Лекционные занятия (16ч.)

1. Базовые концепции кросс-платформенного программирования {дискуссия} (2ч.)[1,2,3,4] Понятие кроссплатформенного программного обеспечения. Проблема

переносимости приложений с одной ОС на другую. Использование библиотек для создания графического интерфейса пользователя (ГИП или в англоязычной нотации GUI graphical user interface) и системных библиотек конкретной операционной системы (платформозависимых библиотек). Особенности кроссплатформенности java. Использование кроссплатформенных библиотек в стандартном языке (C, C++, Perl, Python, Ruby и др.) Веб-программирование как особый подход в кросс платформенном программировании. О кроссплатформенных IDE (Integrated development environment).

2. Особенности программирования для различных операционных систем {дискуссия} (2ч.)[1,2,3,4] Кроссплатформенные среды исполнения. Кроссплатформенные пользовательские интерфейсы. Особенности использования кроссплатформенных библиотек на примере PyQt, PySide. Популярные фреймворки, которые помогут при разработке программ для компьютеров под управлением Windows/MacOS/Linux.

3. Подготовка к программированию на Python с использованием мобильных устройств {дискуссия} (2ч.)[1,2,3,4] Установка и работа в Pydroid 3 на базе Android. Два основных подхода к

работе с интерпретатором Python: непосредственная интерпретация строк кода, вводимых с клавиатуры в интерактивном режиме и выполнение файлов с исходным кодом в пакетном режиме. Вход в интерактивный режим работы. Пакетный режим работы. Python, как язык с неявной сильной динамической типизацией. Отличия динамической и статической типизации, сильной и слабой типизации. Разделение типов данных на встроенные в интерпретатор (built-in) и не встроенные, которые можно использовать при импортировании соответствующих модулей. Основные встроенные типы данных. Объявление и инициализация переменных. Объект, как абстракция для представления данных. Идентификатор как некоторое целочисленное значение, посредством которого уникально адресуется объект. Изменяемые и неизменяемые типы данных.

4. Арифметические операции. Работа со строками в Python {дискуссия} (2ч.)[1,2,3,4] Три встроенных числовых типа данных. Арифметические операции с целыми и вещественными числами. Работа с комплексными числами.

Доступные битовые операции в Python. Представление чисел в других системах счисления. Библиотека math из стандартной поставки Python. Литералы строк. Строки в апострофах и в кавычках. Экранированные последовательности для вставки служебных символов. Использование сырых

строк для подавления экранирования. Строки в тройных апострофах и кавычках. Базовые операции для строк: конкатенация, дублирование строки, определение длины строки, доступ по индексу, извлечение среза.

Форматирование строк с помощью метода `format`.

5. Условные операторы и циклы. Работа со списками {дискуссия} (2ч.)[1,2,3,4]

Конструкция условного оператора ветвления `if`. Использование альтернативного варианта выполнения программы с помощью оператора `if - else`. Реализация выбора из нескольких альтернатив можно с помощью конструкции `if - elif - else`. Оператор цикла `while` для выполнения указанного набора инструкций. Операторы `break` и `continue` для работы с циклами. Оператор `for` для выполнения указанного набора инструкций заданное количество раз. Список (`list`) как структура данных для хранения объектов различных

типов. Хранение списков в памяти. Создание, изменение, удаление списков и работа с его элементами. Создание копий списков и простое присвоение списков. Добавление и удаление элементов из списка. Методы списков.

"Списковое включение" как часть синтаксиса языка для предоставления простого способа построения списков. `List Comprehensions` как обработчик списков. Слайсы (срезы) - составляющая Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. Задание слайса тройкой чисел, разделенных запятой: `start:stop:step`.

6. Кортежи. Словари. Функции в Python {дискуссия} (2ч.)[1,2,3,4]

Кортеж (`tuple`) как неизменяемая структура данных. Причины, по которым стоит использовать кортежи вместо списков. Создание, удаление кортежей и работа с его элементами. Преобразование кортежа в список и обратно. Структура данных, предназначенная для хранения произвольных объектов с доступом по ключу. Хранение данных в формате ключ - значение. Создание, изменение, удаление словарей и работа с его элементами. Проверка наличия ключа в словаре. Методы словарей. Функция как именованный фрагмент программного кода, к которому можно обратиться из другого места программы. Использование ключевого слова `def` для создания функции. Возврат значения функцией. Использование функций для обработки данных. Безымянная функция с произвольным числом аргументов.

7. Работа с исключениями. Ввод-вывод данных. Работа с файлами {дискуссия} (2ч.)[1,2,3,4]

Понятие исключения в языке программирования. Как исключения дают возможность дальнейшей работы в рамках основного алгоритма. Синхронные и асинхронные исключения. Структурная и неструктурная обработка исключений. Синтаксические ошибки. Иерархия исключений в Python.

Обработка исключения внутри синтаксической конструкции `try...except`.

Выполнение определенного программного кода при выходе из блока `try/except` с помощью оператора `finally`. Генерация исключений в Python.

Пользовательские исключения.

Вывод данных в консоль. Функция для считывания вводимых с клавиатуры данных. Преобразование строки в список с помощью метода split() . Считывание списка чисел с одновременным приведением их к типу int . Открытие и закрытие файла. Чтение данных из файла. Запись данных в файл. Дополнительные методы для работы с файлами.

8. Классы и объекты. Итераторы и генераторы {дискуссия} (2ч.)[1,2,3,4] Три основных "столпа" ООП - инкапсуляция, наследование и полиморфизм. Создание класса в Python с помощью инструкции class. Объявление класса, имя класса и тело класса. Создание объекта класса. Статические и динамические атрибуты класса. Методы класса: статические, классовыми (среднее между статическими и обычными) и уровня класса. Конструктор класса и инициализация экземпляра класса. Ключевое слово self. Уровни доступа атрибута и метода в Python. Свойство как метод класса, работа с которым подобна работе с атрибутом. Наследование. Родительский класс. Переопределение методов базового класса в классе наследнике на базе полиморфизма. Инструменты, которые, как правило, используются для поточной обработки данных. Использование итератора для упрощения навигации по элементам объекта. Создание собственных итераторов. Упрощение работы по конструированию итераторов с помощью генераторов.

Лабораторные работы (16ч.)

- 1. Введение в Python(4ч.)[1,2,3,4]** Работа со строками в Python. Условные операторы и циклы. Работа со списками. Кортежи. Словари.
- 2. Работа с файлами. Списки(4ч.)[1,2,3,4]** Функции в Python. Работа с исключениями. Ввод-вывод данных. Работа с файлами. Модули и пакеты. Установка пакетов в Python. Модули и пакеты. Установка пакетов в Python. Итераторы и генераторы.
- 3. Описание объектно-ориентированного подхода(4ч.)[1,2,3,4]** Работа с классами и объектами
- 4. Базы данных(4ч.)[1,2,3,4]** Работа с базой данных SQLite. Основы использования библиотеки PyQt5.

Самостоятельная работа (76ч.)

- 1. Изучение теоретического материала(12ч.)[1,2,3,4]** Изучение конспектов лекций, рекомендованной литературы и интернет-источников
- 2. Подготовка к лабораторным работам(16ч.)[1,2,3,4]** Изучение теоретического материала и подготовка отчета
- 2. Подготовка к лабораторным работам(12ч.)[1,2,3,4]** Изучение теоретического материала и подготовка отчета
- 3. Экзамен(36ч.)[1,2,3,4]** Подготовка к итоговой письменной работе

5. Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине

Для каждого обучающегося обеспечен индивидуальный неограниченный доступ к электронно-библиотечным системам: Лань, Университетская библиотека он-лайн, электронной библиотеке АлтГТУ и к электронной информационно-образовательной среде:

1. Методические указания по выполнению лабораторных работ по дисциплине «Программирование кроссплатформенных систем»

Умбетов С.В. (ИТ)

2023 Учебно-методическое пособие, 680.00 КБ

Дата первичного размещения: 06.06.2023. Обновлено: 06.06.2023.

Прямая

ссылка:

http://elib.altstu.ru/eum/download/it/Umbetov_ProgCrossplSyst_lr_mu.pdf

6. Перечень учебной литературы

6.1. Основная литература

2. Учебное пособие по дисциплине Разработка кроссплатформенных приложений C++ : практикум / составители П. В. Лобзенко, И. В. Щербань. — Ростов-на-Дону : Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. — 71 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/89513.html> (дата обращения: 06.06.2023). — Режим доступа: для авторизир. пользователей

6.2. Дополнительная литература

3. Шелудько, В. М. Язык программирования высокого уровня Python : функции, структуры данных, дополнительные модули : учебное пособие : [16+] / В. М. Шелудько. — Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. — 108 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=500060> (дата обращения: 06.06.2023). — Библиогр. в кн. — ISBN 978-5-9275-2648-2. — Текст : электронный.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

4. <https://pythonworld.ru/>

8. Фонд оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации

Содержание промежуточной аттестации раскрывается в комплекте

контролирующих материалов, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС, которые хранятся на кафедре-разработчике РПД в печатном виде и в ЭИОС.

Фонд оценочных материалов (ФОМ) по дисциплине представлен в приложении А.

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

Для успешного освоения дисциплины используются ресурсы электронной информационно-образовательной среды, образовательные интернет-порталы, глобальная компьютерная сеть Интернет. В процессе изучения дисциплины происходит интерактивное взаимодействие обучающегося с преподавателем через личный кабинет студента.

№пп	Используемое программное обеспечение
1	Android Studio
2	Chrome
3	LibreOffice
4	PyCharm Community Edition
5	Python
6	Windows
7	Антивирус Kaspersky

№пп	Используемые профессиональные базы данных и информационные справочные системы
1	Бесплатная электронная библиотека онлайн "Единое окно к образовательным ресурсам" для студентов и преподавателей; каталог ссылок на образовательные интернет-ресурсы (http://Window.edu.ru)
2	Национальная электронная библиотека (НЭБ) — свободный доступ читателей к фондам российских библиотек. Содержит коллекции оцифрованных документов (как открытого доступа, так и ограниченных авторским правом), а также каталог изданий, хранящихся в библиотеках России. (http://нэб.рф/)

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Наименование специальных помещений и помещений для самостоятельной работы
учебные аудитории для проведения учебных занятий
помещения для самостоятельной работы

Материально-техническое обеспечение и организация образовательного процесса по дисциплине для инвалидов и лиц с ограниченными возможностями здоровья осуществляется в соответствии с «Положением об обучении инвалидов и лиц с ограниченными возможностями здоровья».