

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Порохня Андрей Александрович

Должность: И.о. директора института перспективной инженерии

Дата подписания: 19.06.2026 15:26:10

Уникальный программный ключ:

990bea3aeec48c23bd8699e48288f8d1960c600

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

И.о. директора института
перспективной инженерии,
канд. техн. наук, доцент
А.А. Порохня

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
«Основы кроссплатформенного программирования»

Направление подготовки	11.03.02 Инфокоммуникационные технологии и системы связи
Направленность (профиль)	Интеллектуальная обработка данных в инфокоммуникационных системах и сетях
Год начала обучения	2026 г.
Форма обучения	очно-заочная
Реализуется в семестре	3

Введение

1. Назначение

Фонд оценочных средств по дисциплине «Основы кроссплатформенного программирования» предназначен для объективной оценки уровня сформированности компетенций.

2. ФОС является приложением к программе дисциплины «Основы кроссплатформенного программирования».

3. Разработчик Р. А. Воронкин, доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии.

4. Проведена экспертиза ФОС.

Члены экспертной группы:

Председатель: Н.Ю. Братченко, председатель УМК института перспективной инженерии, доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии.

Члены комиссии:

В.П. Мочалов, профессор департамента цифровых, робототехнических систем и электроники института перспективной инженерии.

С.В. Мельников, старший научный сотрудник междисциплинарного учебно-исследовательского центра агротехнологий, доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии.

Представитель организации работодателя:

Миронов В.А., генеральный директор ООО «Оринтекс».

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

5.Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Компетенция (ии), индикатор (ы)	Уровни сформированности компетенции(ий),			
	Минимальный уровень не достигнут (Неудовлетвори тельно) 2 балла	Минимальный уровень (удовлетворите льно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-6				
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ИД-1 _{ПК-6} понимает архитектуру и общие принципы функционирования, аппаратных, программных и программно-аппаратных средств администрируемой сети, порядок настройки сетевых элементов инфокоммуникационной системы.	Не понимает общие принципы функционирования аппаратных, программных и программно-аппаратных средств, не различает особенности программной среды разных платформ, не способен объяснить порядок настройки элементов программной и системной среды.	Обладает фрагментарным представлением об архитектуре и принципах функционирования аппаратных, программных и программно-аппаратных средств, в общих чертах различает особенности программной среды различных платформ, но допускает ошибки при объяснении порядка настройки системных и программных компонентов.	Понимает архитектуру и общие принципы функционирования аппаратных, программных и программно-аппаратных средств, учитывает основные различия программной среды различных платформ, в целом правильно объясняет порядок настройки системных и программных компонентов, необходимых для функционирования кроссплатформенных приложений.	Уверенно понимает архитектуру и общие принципы функционирования аппаратных, программных и программно-аппаратных средств, корректно объясняет особенности программной среды различных платформ, обоснованно описывает порядок настройки системных и программных компонентов, необходимых для разработки и функционирования кроссплатформенных приложений.
Результаты обучения по дисциплине (модулю): <i>Индикатор:</i> ИД-5 _{ПК-6} выполняет установку и настройку системного и прикладного ПО, необходимого для функционирования ИС в соответствии с трудовым заданием, осуществляет типовое	Не способен выполнить установку и настройку системного и прикладного программного обеспечения, необходимого для функционирования информационной системы. Не владеет базовыми приемами параметризации	Выполняет установку и настройку системного и прикладного программного обеспечения только с существенными затруднениями и ошибками. Осуществляет параметризацию программной среды на базовом	В целом правильно выполняет установку и настройку системного и прикладного программного обеспечения, необходимого для функционирования информационной системы. Осуществляет	Уверенно выполняет установку и настройку системного и прикладного программного обеспечения, необходимого для функционирования информационной системы.

проектирование и параметризацию ИС в соответствии с регламентами организации.	программной среды и не ориентируется в типовых процедурах настройки.	уровне, но не всегда учитывает требования задания и установленные регламенты.	типовое проектирование и параметризацию программной среды с учетом основных требований задания и регламентов организации, допуская незначительные неточности.	Грамотно осуществляет типовое проектирование и параметризацию программной среды в соответствии с трудовым заданием и регламентами организации, обеспечивая корректное функционирование программного решения.
---	--	---	---	--

Оценивание уровня сформированности компетенции по дисциплине осуществляется на основе «Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры - в федеральном государственном автономном образовательном учреждении высшего образования «Северо-Кавказский федеральный университет» в актуальной редакции.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Номер задания	Правильный ответ	Содержание вопроса	Компетенция
		Форма обучения очная, Семестр 3 (3)	
1	Кроссплатформенное программирование - это подход к разработке программного обеспечения, при котором приложение может функционировать в разных операционных системах с минимальными изменениями или без них.	Дайте определение понятию «кроссплатформенное программирование».	ПК-6
2	Python	Какой язык программирования широко используется для разработки переносимых приложений благодаря наличию интерпретатора и богатой стандартной библиотеки?	ПК-6
3	В) pathlib	Какой модуль Python рекомендуется использовать для кроссплатформенной работы с путями? А) math Б) random В) pathlib Г) threading	ПК-6
4	Операционная система, файловая система, переменные окружения, системные пути, способ запуска процессов.	Назовите основные платформенно-зависимые аспекты, которые необходимо учитывать при разработке кроссплатформенных программ.	ПК-6
5	1. Установка интерпретатора; 2. Создание виртуального окружения; 3. Установка зависимостей; 4. Настройка параметров среды; 5. Запуск и проверка работы приложения.	Упорядочите этапы подготовки среды для запуска кроссплатформенного Python-приложения: установка зависимостей, запуск и проверка работы приложения, создание виртуального окружения, установка интерпретатора, настройка параметров среды.	ПК-6
6	Переменные окружения - это именованные значения, задаваемые в операционной системе и используемые	Что такое переменные окружения в контексте кроссплатформенного программирования?	ПК-6

	программами для получения параметров своей работы.		
7	Г) Все перечисленное	Какие задачи можно решать средствами кроссплатформенного программирования на Python? А) Работа с файлами Б) Запуск внешних процессов В) Получение сведений о платформе Г) Все перечисленное	ПК-6
8	Виртуальное окружение - это изолированная среда Python, предназначенная для установки и использования зависимостей конкретного проекта отдельно от системной установки Python.	Что такое виртуальное окружение Python?	ПК-6
9	Б) platform	Какой модуль Python позволяет получить сведения об операционной системе и архитектуре платформы? А) os.path Б) platform В) json Г) typing	ПК-6
10	Для обеспечения переносимости программного обеспечения, корректной настройки среды выполнения, уменьшения числа ошибок совместимости и упрощения сопровождения приложения.	С какой целью при разработке кроссплатформенного приложения учитываются особенности различных операционных систем и программных сред?	ПК-6
11	Кроссплатформенность позволяет использовать одну программную разработку в разных операционных системах, снижая затраты на сопровождение и доработку.	В чем состоит основное преимущество кроссплатформенного программирования?	ПК-6
12	Б) os	Какой модуль Python предоставляет средства для взаимодействия с операционной системой? А) time Б) os В) re Г) csv	ПК-6
13	Разделитель пути в Windows обычно \, а в Linux и macOS обычно /.	В чем состоит различие представления путей к файлам в Windows и Linux?	ПК-6

14	Использование относительных путей, модуля pathlib, виртуальных окружений и стандартной библиотеки Python.	Назовите основные средства обеспечения переносимости Python-приложения.	ПК-6
15	Б) sys.argv	С помощью какого средства Python можно получить аргументы командной строки? А) os.environ Б) sys.argv В) platform.system() Г) input()	ПК-6
16	Проверить установленную версию Python, создать виртуальное окружение, активировать его, установить необходимые зависимости.	Какие действия необходимо выполнить перед началом работы над кроссплатформенным Python-проектом?	ПК-6
17	os.environ	Какой объект Python используется для получения доступа к переменным окружения?	ПК-6
18	Виртуальные окружения позволяют изолировать зависимости проекта и избежать конфликтов между версиями библиотек.	Для чего в кроссплатформенной разработке используются виртуальные окружения?	ПК-6
19	В) subprocess	Какой модуль Python применяется для запуска внешних процессов? А) argparse Б) shutil В) subprocess Г) logging	ПК-6
20	Потому что разные операционные системы могут по-разному обрабатывать пути, команды, переменные окружения и системные вызовы.	Почему при разработке переносимого приложения необходимо учитывать особенности операционной системы?	ПК-6
21	platform.system()	Какая функция Python позволяет определить имя текущей операционной системы?	ПК-6
22	А) pathlib.Path	Какой объект рекомендуется использовать для безопасной и переносимой работы с путями? А) pathlib.Path Б) open() В) enumerate() Г) set()	ПК-6
23	Создание каталогов, проверка существования файлов, чтение и запись данных, получение списка файлов.	Какие типовые операции с файловой системой выполняются в кроссплатформенных приложениях?	ПК-6

24	Настройка среды включает установку интерпретатора, библиотек, переменных окружения и параметров запуска программы.	Что понимается под настройкой программной среды для функционирования информационной системы?	ПК-6
25	Б) Относительные пути	Какие пути обычно являются более переносимыми в кроссплатформенном приложении? А) Абсолютные системные пути Б) Относительные пути В) Только сетевые пути Г) Только временные пути	ПК-6
26	Проверка работоспособности программы на разных операционных системах и в разных условиях запуска.	Что является основной целью тестирования кроссплатформенного приложения?	ПК-6
27	Модульность позволяет отделить платформенно-зависимые части программы от основной логики и упростить сопровождение.	Какую роль играет модульность в разработке переносимого программного обеспечения?	ПК-6
28	Г) Все перечисленное	Какие характеристики важны для кроссплатформенного приложения? А) Переносимость Б) Сопровождаемость В) Корректная работа в разных ОС Г) Все перечисленное	ПК-6
29	Использование параметров конфигурации позволяет адаптировать поведение программы к различным условиям эксплуатации без изменения исходного кода.	Зачем в кроссплатформенных приложениях используется параметризация?	ПК-6
30	Последовательность: установка Python, создание виртуального окружения, установка библиотек, настройка параметров, тестирование запуска.	Установите правильную последовательность подготовки среды для запуска Python-приложения на новой платформе.	ПК-6
31	Кроссплатформенное приложение должно учитывать различия операционных систем, но сохранять единый основной программный код.	В чем состоит сущность разработки кроссплатформенного приложения?	ПК-6

32	В) sys	Какой модуль Python используется для получения сведений о параметрах запуска программы и аргументах командной строки? А) os Б) platform В) sys Г) shutil	ПК-6
33	Использование стандартной библиотеки Python позволяет сократить зависимость программы от конкретной платформы и упростить переносимость.	Почему при разработке переносимых приложений рекомендуется использовать стандартную библиотеку Python?	ПК-6
34	Path.exists()	Какой метод объекта Path используется для проверки существования файла или каталога?	ПК-6
35	Потому что абсолютные пути зависят от конкретной структуры каталогов операционной системы и снижают переносимость программы.	Почему в кроссплатформенных приложениях не рекомендуется жестко задавать абсолютные пути?	ПК-6
36	Б) os.getenv()	Какая функция Python позволяет получить значение переменной окружения? А) sys.get() Б) os.getenv() В) platform.getenv() Г) env.read()	ПК-6
37	Создание, настройка и использование изолированной программной среды для конкретного проекта.	Что является основной задачей виртуального окружения в Python-проекте?	ПК-6
38	platform.machine()	Какая функция Python может использоваться для получения сведений об архитектуре машины?	ПК-6
39	Платформенно-зависимая логика связана с особенностями конкретной ОС, а платформенно-независимая реализует общую функциональность программы.	В чем различие между платформенно-зависимой и платформенно-независимой логикой приложения?	ПК-6
40	А) Проверка работы программы в разных операционных системах	Что в наибольшей степени характеризует тестирование кроссплатформенного приложения? А) Проверка работы программы в разных операционных системах Б) Только проверка синтаксиса кода В) Только анализ скорости работы программы Г) Только проверка интерфейса	ПК-6

41	Модуль subprocess позволяет запускать внешние программы и управлять их выполнением из Python-приложения.	Для чего используется модуль subprocess в кроссплатформенном программировании?	ПК-6
42	Б) pathlib и os	Какие модули Python чаще всего используются для работы с файловой системой? А) json и csv Б) pathlib и os В) math и statistics Г) threading и asyncio	ПК-6
43	Проверка существования файла, чтение и запись данных, создание каталогов, формирование корректных путей.	Какие действия должен уметь выполнять разработчик при работе с файловой системой в переносимом приложении?	ПК-6
44	Параметризация позволяет изменять поведение программы через настройки, не изменяя ее исходный код.	Что понимается под параметризацией программного обеспечения?	ПК-6
45	В) Виртуальное окружение	Что необходимо создать для изоляции библиотек конкретного Python-проекта? А) Отдельный системный раздел Б) Новый каталог пользователя В) Виртуальное окружение Г) Архив резервной копии	ПК-6
46	Использование переносимых библиотек, отказ от жестко заданных путей, учет различий ОС, тестирование на нескольких платформах.	Назовите основные способы повышения переносимости программного обеспечения.	ПК-6
47	Path.mkdir()	Какой метод объекта Path используется для создания каталога?	ПК-6
48	Потому что одинаковый программный код может вести себя по-разному в зависимости от операционной системы и условий выполнения.	Почему важно проверять кроссплатформенное приложение в разных средах выполнения?	ПК-6
49	Г) Все перечисленное	Что может входить в настройку программной среды кроссплатформенного приложения? А) Установка зависимостей Б) Настройка переменных окружения В) Подготовка виртуального окружения Г) Все перечисленное	ПК-6
50	Последовательность действий: определить платформенные	Установите правильную последовательность этапов разработки кроссплатформенного приложения.	ПК-6

	особенности, выбрать переносимые средства реализации, настроить среду, реализовать функциональность, выполнить тестирование.		
51	Кроссплатформенность обеспечивает возможность использования программного продукта в различных операционных системах без существенной переработки исходного кода.	Что понимается под кроссплатформенностью программного обеспечения?	ПК-6
52	Б) platform.platform()	Какая функция Python позволяет получить строковое описание текущей платформы? А) os.platform() Б) platform.platform() В) sys.platform() Г) pathlib.platform()	ПК-6
53	Использование модуля pathlib позволяет формировать пути к файлам и каталогам в переносимой форме, не привязываясь к конкретному разделителю пути.	Почему для работы с путями в кроссплатформенных приложениях рекомендуется использовать pathlib?	ПК-6
54	sys.platform	Какой атрибут Python позволяет получить краткий идентификатор текущей платформы?	ПК-6
55	Потому что команды, пути, переменные окружения и поведение системных средств могут различаться в зависимости от операционной системы.	Почему запуск внешних процессов требует учета платформенной специфики?	ПК-6
56	В) Path.home()	Какой метод объекта Path используется для получения домашнего каталога пользователя? А) Path.root() Б) Path.user() В) Path.home() Г) Path.main()	ПК-6
57	Настройка среды разработки включает установку интерпретатора, выбор редактора или IDE, создание виртуального окружения и подключение необходимых библиотек.	Что включает в себя базовая настройка среды разработки для кроссплатформенного Python-проекта?	ПК-6

58	Использование относительных путей, стандартной библиотеки, виртуальных окружений и параметров конфигурации.	Перечислите основные средства, повышающие переносимость Python-приложения.	ПК-6
59	А) subprocess.run()	Какое средство Python удобно использовать для простого запуска внешней команды? А) subprocess.run() Б) platform.run() В) sys.exec() Г) pathlib.run()	ПК-6
60	Модульная структура упрощает сопровождение, тестирование и адаптацию программы к различным платформам.	Какое значение имеет модульная организация кода в кроссплатформенном программировании?	ПК-6
61	Path.cwd()	Какой метод объекта Path используется для получения текущего рабочего каталога?	ПК-6
62	Б) os.name	Какой атрибут Python позволяет получить обобщенное имя используемой операционной системы? А) platform.id Б) os.name В) sys.name Г) env.name	ПК-6
63	Проверка должна проводиться на разных платформах, чтобы выявить различия в поведении программы и ошибки совместимости.	В чем состоит назначение тестирования кроссплатформенного приложения на нескольких операционных системах?	ПК-6
64	Параметры конфигурации позволяют адаптировать программу к условиям эксплуатации без изменения ее исходного кода.	Для чего в кроссплатформенном приложении используются конфигурационные параметры?	ПК-6
65	Г) В разных операционных системах используются разные способы представления путей и доступа к ресурсам	Почему нельзя без проверки считать, что файловые операции будут одинаково работать на всех платформах? А) Потому что Python не поддерживает файлы Б) Потому что в Linux нельзя работать с каталогами В) Потому что все платформы используют одинаковые правила Г) В разных операционных системах используются разные способы представления путей и доступа к ресурсам	ПК-6

66	Path.open()	Какой метод объекта Path можно использовать для открытия файла?	ПК-6
67	Установка зависимостей в виртуальное окружение позволяет изолировать проект и обеспечить воспроизводимость его программной среды.	Почему рекомендуется устанавливать зависимости проекта в виртуальное окружение?	ПК-6
68	В) os.environ	Какое средство Python используется для доступа ко всем переменным окружения в виде отображения? А) sys.env Б) platform.environ В) os.environ Г) pathlib.environ	ПК-6
69	К типовым этапам относятся установка Python, создание виртуального окружения, установка библиотек, настройка параметров среды и проверка запуска программы.	Назовите основные этапы подготовки программной среды для работы кроссплатформенного приложения.	ПК-6
70	Последовательность: анализ платформенных требований, выбор переносимых средств реализации, настройка программной среды, разработка приложения, тестирование на разных платформах.	Установите правильную последовательность основных этапов разработки и подготовки кроссплатформенного программного решения.	ПК-6
71	Кроссплатформенное программирование предполагает разработку программ, способных работать в различных операционных системах при сохранении единой логики и минимальной зависимости от платформы.	Дайте определение кроссплатформенного программирования.	ПК-6
72	А) Path	Какой класс из модуля pathlib предназначен для представления путей к файлам и каталогам? А) Path Б) File В) Directory Г) Route	ПК-6

73	Использование стандартных библиотек и переносимых средств разработки позволяет уменьшить зависимость программы от особенностей конкретной операционной системы.	Почему в кроссплатформенной разработке важно использовать переносимые программные средства?	ПК-6
74	platform.system()	Какая функция Python позволяет определить используемую операционную систему?	ПК-6
75	Потому что разные операционные системы используют различные разделители путей, правила именования файлов и способы доступа к системным ресурсам.	Почему при разработке кроссплатформенных приложений необходимо учитывать особенности файловой системы?	ПК-6
76	Б) Path.read_text()	Какой метод объекта Path позволяет прочитать текстовое содержимое файла? А) Path.get_text() Б) Path.read_text() В) Path.load_text() Г) Path.text()	ПК-6
77	Под настройкой программной среды понимается подготовка интерпретатора, библиотек, переменных окружения и параметров запуска, необходимых для корректной работы приложения.	Что понимается под настройкой программной среды в кроссплатформенном программировании?	ПК-6
78	Использование виртуального окружения, установка зависимостей, настройка параметров среды и проверка запуска приложения.	Какие действия включает первичная подготовка среды выполнения Python-приложения?	ПК-6
79	В) sys.executable	Какое средство Python позволяет получить путь к текущему исполняемому файлу интерпретатора? А) platform.python() Б) os.python В) sys.executable Г) pathlib.python	ПК-6
80	Модульность способствует разделению программы на независимые части, упрощает сопровождение и позволяет изолировать платформенно-зависимые компоненты.	Какое значение имеет модульность при разработке кроссплатформенного программного обеспечения?	ПК-6

81	os.getenv()	Какая функция Python используется для получения значения отдельной переменной окружения?	ПК-6
82	Г) Проверка работы программы в нескольких операционных системах	Что является важнейшим признаком тестирования кроссплатформенного приложения? А) Проверка только синтаксиса Б) Проверка только быстродействия В) Проверка только интерфейса Г) Проверка работы программы в нескольких операционных системах	ПК-6
83	Переносимость программного обеспечения достигается за счет учета платформенных различий, использования стандартных библиотек и отказа от жесткой привязки к конкретной системе.	За счет чего обеспечивается переносимость программного обеспечения?	ПК-6
84	Path.write_text()	Какой метод объекта Path позволяет записать текст в файл?	ПК-6
85	Потому что запуск внешних процессов может зависеть от формата команд, доступных утилит и особенностей системной оболочки конкретной операционной системы.	Почему запуск внешних команд в кроссплатформенном приложении требует дополнительной проверки?	ПК-6
86	А) subprocess	Какой модуль Python используется для запуска и управления внешними процессами? А) subprocess Б) platform В) pathlib Г) venv	ПК-6
87	Параметризация позволяет изменять режимы работы программы и адаптировать ее поведение к условиям эксплуатации без изменения исходного кода.	Какую роль играет параметризация в кроссплатформенном приложении?	ПК-6
88	К типовым средствам относятся виртуальные окружения, конфигурационные файлы, стандартная библиотека Python и переносимые	Назовите типовые средства, применяемые при разработке кроссплатформенных Python-приложений.	ПК-6

	способы работы с файлами и процессами.		
89	Б) venv	Какой модуль стандартной библиотеки Python используется для создания виртуальных окружений? А) env Б) venv В) virtual Г) pipenv	ПК-6
90	Последовательность: установка интерпретатора, создание виртуального окружения, установка библиотек, настройка параметров среды, тестирование работы приложения.	Установите правильную последовательность подготовки кроссплатформенного Python-приложения к запуску.	ПК-6
91	Кроссплатформенное приложение должно обеспечивать корректную работу в различных операционных системах за счет использования переносимых средств реализации и учета платформенных различий.	Что характеризует кроссплатформенное приложение?	ПК-6
92	В) Path.is_file()	Какой метод объекта Path используется для проверки того, что путь указывает на файл? А) Path.check_file() Б) Path.file_exists() В) Path.is_file() Г) Path.has_file()	ПК-6
93	Использование переносимых средств разработки позволяет уменьшить зависимость программы от конкретной операционной системы и повысить удобство сопровождения.	Почему в кроссплатформенном программировании важно использовать переносимые средства реализации?	ПК-6
94	Path.is_dir()	Какой метод объекта Path используется для проверки того, что путь указывает на каталог?	ПК-6
95	Потому что в разных операционных системах различаются правила обработки путей, структура каталогов и способы обращения к системным ресурсам.	Почему при работе с файлами и каталогами необходимо учитывать особенности платформы?	ПК-6

96	A) platform.machine()	Какая функция Python позволяет получить сведения об архитектуре используемой машины? А) platform.machine() Б) sys.machine() В) os.machine() Г) pathlib.machine()	ПК-6
97	Проверка существования файла, создание каталогов, чтение и запись данных, формирование переносимых путей.	Какие базовые действия с файловой системой должен уметь выполнять разработчик кроссплатформенного приложения?	ПК-6
98	Использование переменных окружения позволяет передавать приложению параметры выполнения и учитывать особенности программной среды.	Для чего в кроссплатформенном приложении используются переменные окружения?	ПК-6
99	Б) sys.platform	Какое средство Python позволяет получить идентификатор текущей платформы? А) platform.name Б) sys.platform В) os.platform_name Г) env.platform	ПК-6
100	Виртуальное окружение позволяет изолировать зависимости проекта, обеспечить воспроизводимость среды и упростить сопровождение приложения.	В чем состоит назначение виртуального окружения в Python-проекте?	ПК-6
101	Path.iterdir()	Какой метод объекта Path используется для перебора содержимого каталога?	ПК-6
102	Г) Все перечисленное	Что может входить в подготовку программной среды кроссплатформенного приложения? А) Установка интерпретатора Б) Установка библиотек В) Настройка переменных окружения Г) Все перечисленное	ПК-6
103	Тестирование на нескольких платформах позволяет выявить ошибки совместимости и различия в поведении программы в разных средах выполнения.	В чем состоит назначение тестирования кроссплатформенного приложения в разных операционных системах?	ПК-6

104	Path.unlink()	Какой метод объекта Path используется для удаления файла?	ПК-6
105	Потому что системные команды, оболочки и доступные утилиты могут различаться в зависимости от операционной системы.	Почему при запуске внешних процессов необходимо учитывать платформенную специфику?	ПК-6
106	В) subprocess.Popen()	Какое средство Python используется для более гибкого управления внешним процессом? А) os.process() Б) sys.run() В) subprocess.Popen() Г) platform.exec()	ПК-6
107	Модульная организация кода упрощает сопровождение, тестирование и выделение платформенно-зависимых частей программы.	Какую роль играет модульность при разработке переносимого программного обеспечения?	ПК-6
108	Использование конфигурационных параметров позволяет адаптировать поведение программы к условиям эксплуатации без изменения исходного кода.	В чем заключается значение параметризации в кроссплатформенном приложении?	ПК-6
109	Б) Path.resolve()	Какой метод объекта Path используется для получения абсолютного пути? А) Path.absolute_path() Б) Path.resolve() В) Path.fullname() Г) Path.abs()	ПК-6
110	Последовательность: анализ требований среды, выбор переносимых средств реализации, настройка программной среды, разработка решения, тестирование на разных платформах.	Установите правильную последовательность основных этапов разработки кроссплатформенного программного решения.	ПК-6
111	Кроссплатформенное программирование предполагает создание программного обеспечения, способного функционировать в различных операционных системах при	Что понимается под кроссплатформенным программированием?	ПК-6

	минимальной зависимости от платформы.		
112	A) Path.exists()	Какой метод объекта Path используется для проверки существования файла или каталога? A) Path.exists() Б) Path.check() В) Path.find() Г) Path.present()	ПК-6
113	Использование стандартной библиотеки Python позволяет разрабатывать более переносимые приложения и уменьшать зависимость от особенностей конкретной платформы.	Почему в кроссплатформенной разработке рекомендуется использовать стандартную библиотеку Python?	ПК-6
114	platform.version()	Какая функция Python может использоваться для получения информации о версии платформы?	ПК-6
115	Потому что в разных операционных системах различаются соглашения об именовании файлов, структура каталогов и механизмы обращения к системным ресурсам.	Почему при разработке переносимого приложения важно учитывать особенности файловой системы?	ПК-6
116	В) Path.parent	Какое свойство объекта Path позволяет получить родительский каталог? А) Path.rootdir Б) Path.prev В) Path.parent Г) Path.up()	ПК-6
117	Первичная настройка среды включает установку интерпретатора Python, создание виртуального окружения, установку необходимых библиотек и настройку параметров запуска.	Что включает первичная подготовка среды для работы с кроссплатформенным Python-приложением?	ПК-6
118	Использование относительных путей, виртуальных окружений, параметров конфигурации и переносимых модулей стандартной библиотеки.	Перечислите основные средства обеспечения переносимости Python-приложения.	ПК-6
119	Б) os.chdir()	Какая функция Python используется для изменения текущего рабочего каталога? А) sys.chdir() Б) os.chdir() В) pathlib.chdir() Г) platform.chdir()	ПК-6

120	Модульная организация программы позволяет отделить платформенно-зависимые части от общей логики и облегчает сопровождение программного обеспечения.	В чем состоит значение модульной организации кода в кроссплатформенном программировании?	ПК-6
121	Path.name	Какое свойство объекта Path позволяет получить имя файла или каталога без пути?	ПК-6
122	А) Тестирование программы в нескольких операционных системах	Что наиболее полно характеризует тестирование кроссплатформенного приложения? А) Тестирование программы в нескольких операционных системах Б) Только синтаксическая проверка кода В) Только проверка скорости выполнения Г) Только проверка интерфейса	ПК-6
123	Переносимость программного обеспечения обеспечивается применением стандартных средств разработки, учетом платформенных различий и отказом от жесткой привязки к конкретной среде.	За счет каких подходов обеспечивается переносимость программного обеспечения?	ПК-6
124	Path.suffix	Какое свойство объекта Path позволяет получить расширение файла?	ПК-6
125	Потому что набор доступных команд, синтаксис их вызова и поведение системной оболочки могут различаться на разных платформах.	Почему запуск внешних команд в кроссплатформенном приложении требует учета особенностей операционной системы?	ПК-6
126	Г) platform.python_version()	Какая функция Python позволяет получить версию используемого интерпретатора Python? А) sys.python_version() Б) os.python_version() В) venv.version() Г) platform.python_version()	ПК-6
127	Конфигурационные параметры позволяют адаптировать работу приложения к различным условиям	Какую роль играет параметризация в кроссплатформенном программном обеспечении?	ПК-6

	эксплуатации без изменения исходного кода программы.		
128	Типовые средства включают виртуальные окружения, конфигурационные файлы, стандартную библиотеку Python, а также переносимые средства работы с файлами, каталогами и процессами.	Назовите типовые средства, применяемые при разработке кроссплатформенных приложений на Python.	ПК-6
129	В) Path.stem	Какое свойство объекта Path позволяет получить имя файла без расширения? А) Path.file Б) Path.basename В) Path.stem Г) Path.title	ПК-6
130	Последовательность: установка Python, создание виртуального окружения, установка зависимостей, настройка параметров среды, тестирование запуска приложения.	Установите правильную последовательность подготовки кроссплатформенного Python-приложения к запуску.	ПК-6
131	Кроссплатформенное программирование ориентировано на создание программ, способных корректно работать в различных операционных системах при минимальных изменениях исходного кода.	В чем заключается сущность кроссплатформенного программирования?	ПК-6
132	Б) Path.cwd()	Какой метод класса Path используется для получения текущего рабочего каталога? А) Path.current() Б) Path.cwd() В) Path.workdir() Г) Path.here()	ПК-6
133	Использование переносимых средств разработки и стандартной библиотеки Python позволяет снизить зависимость программы от особенностей конкретной операционной системы.	Почему при разработке кроссплатформенного приложения важно использовать переносимые средства реализации?	ПК-6
134	os.getcwd()	Какая функция Python позволяет получить текущий рабочий каталог?	ПК-6

135	Потому что разные операционные системы используют различные соглашения о путях, правах доступа и организации файловой системы.	Почему при работе с файлами в кроссплатформенном приложении необходимо учитывать платформенные различия?	ПК-6
136	A) Path.touch()	Какой метод объекта Path используется для создания пустого файла? A) Path.touch() Б) Path.create() В) Path.makefile() Г) Path.new()	ПК-6
137	Подготовка программной среды включает установку интерпретатора, создание виртуального окружения, установку библиотек и настройку параметров выполнения программы.	Что включает подготовка программной среды для кроссплатформенного Python-приложения?	ПК-6
138	Использование виртуального окружения позволяет изолировать зависимости проекта и обеспечить воспроизводимость среды выполнения.	Для чего при разработке кроссплатформенного Python-приложения используется виртуальное окружение?	ПК-6
139	B) os.listdir()	Какая функция Python используется для получения списка имен файлов и каталогов в указанной директории? A) sys.listdir() Б) platform.listdir() В) os.listdir() Г) pathlib.listdir()	ПК-6
140	Модульная организация кода позволяет отделить платформенно-зависимые части программы от общей логики и облегчает сопровождение приложения.	В чем состоит значение модульной организации программы в кроссплатформенной разработке?	ПК-6
141	Path.mkdir()	Какой метод объекта Path применяется для создания каталога?	ПК-6
142	Г) Проверка корректности работы программы в разных средах выполнения	Что является основной задачей тестирования кроссплатформенного приложения? A) Проверка только синтаксиса Б) Проверка только оформления кода В) Проверка только скорости работы Г) Проверка корректности работы программы в разных средах выполнения	ПК-6

143	Переносимость достигается за счет использования стандартных библиотек, отказа от жесткой платформенной привязки и тестирования в разных операционных системах.	За счет каких мер обеспечивается переносимость программного обеспечения?	ПК-6
144	os.environ	Какое средство Python позволяет получить доступ к переменным окружения текущего процесса?	ПК-6
145	Потому что формат команд, доступные утилиты и поведение системной оболочки могут различаться в зависимости от операционной системы.	Почему запуск внешних процессов требует учета особенностей конкретной платформы?	ПК-6
146	Б) subprocess.run()	Какое средство Python рекомендуется использовать для запуска внешней команды с ожиданием завершения? А) os.run() Б) subprocess.run() В) sys.process() Г) platform.run()	ПК-6
147	Параметризация позволяет адаптировать поведение программы к условиям эксплуатации без изменения исходного кода.	Какова роль параметризации в кроссплатформенном приложении?	ПК-6
148	К типовым средствам относятся стандартная библиотека Python, виртуальные окружения, конфигурационные файлы и переносимые способы работы с файлами, каталогами и процессами.	Назовите типовые средства, используемые при разработке кроссплатформенных Python-приложений.	ПК-6
149	А) Path.glob()	Какой метод объекта Path используется для поиска файлов по шаблону? А) Path.glob() Б) Path.search() В) Path.find() Г) Path.pattern()	ПК-6
150	Последовательность: анализ требований среды, выбор переносимых средств реализации, подготовка программной среды, разработка	Установите правильную последовательность этапов разработки кроссплатформенного программного решения.	ПК-6

	приложения, тестирование на различных платформах.		
--	---	--	--

2. Описание шкалы оценивания

Успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации.

3. Критерии оценивания компетенций*

3.1. Оценка компетенции ПК-6

Компетенция ПК-6 направлена на формирование у обучающихся знаний и практических умений, связанных с пониманием принципов функционирования программной среды, особенностей различных платформ и использованием средств разработки, настройки и сопровождения кроссплатформенного программного обеспечения. В рамках дисциплины «Основы кроссплатформенного программирования» ее формирование связано с освоением базовых подходов к созданию переносимых программных решений на Python, учетом особенностей операционных систем, использованием стандартной библиотеки, а также выполнением заданий по настройке программной среды и разработке программ, корректно работающих в различных условиях эксплуатации.

Критерии достижения ПК-6 включают:

- знание сущности кроссплатформенного программирования, его целей, задач и области применения;
- понимание архитектуры и общих принципов функционирования аппаратных, программных и программно-аппаратных средств, влияющих на выполнение программного обеспечения;
- знание особенностей программной среды различных операционных систем и факторов, влияющих на переносимость приложений;
- понимание принципов использования языка Python для разработки переносимых программных решений;
- знание базовых средств стандартной библиотеки Python для работы с файловой системой, путями, переменными окружения, процессами и сведениями о платформе;
- умение выполнять установку и настройку системного и прикладного программного обеспечения, необходимого для функционирования программного решения;
- умение использовать виртуальные окружения, зависимости и параметры конфигурации при подготовке программной среды;
- способность применять типовые подходы к проектированию и параметризации кроссплатформенных приложений;
- умение учитывать платформенные различия при разработке, запуске и тестировании программного обеспечения;

- способность анализировать условия эксплуатации приложения и обеспечивать корректное функционирование программного решения на различных платформах.

Оценка осуществляется через теоретические вопросы, тестирование, собеседование, контрольные работы, защиту лабораторных работ и проектов, связанных с разработкой, настройкой и проверкой работоспособности кроссплатформенных программных приложений.

3.2. Оценка уровня сформированности

Уровень достижения компетенций оценивается по четырёхбалльной шкале:

- **2 (низкий)** — компетенция не сформирована, действия фрагментарны или некорректны;
- **3 (достаточный)** — компетенция сформирована на базовом уровне, имеются неточности;
- **4 (хороший)** — уверенное применение знаний и умений, допущены незначительные ошибки;
- **5 (высокий)** — компетенция сформирована в полном объёме, решения обоснованы, результаты устойчивы.

Компетенции ПК-1 и ПК-2 развиваются в тесной связке, обеспечивая подготовку обучающихся к решению реальных задач в области анализа данных и интеллектуальных технологий.