

ПРИЛОЖЕНИЕ А
ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ «Программирование кроссплатформенных систем»

1. Перечень оценочных средств для компетенций, формируемых в результате освоения дисциплины

Код контролируемой компетенции	Способ оценивания	Оценочное средство
ПК-12: Способен разрабатывать программы и их блоки, проводить их отладку и настройку для построения интеллектуальных систем и приборов	Экзамен	Комплект контролирующих материалов для экзамена

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Оцениваемые компетенции представлены в разделе «Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций» рабочей программы дисциплины «Программирование кроссплатформенных систем».

При оценивании сформированности компетенций по дисциплине «Программирование кроссплатформенных систем» используется 100-балльная шкала.

Критерий	Оценка по 100-балльной шкале	Оценка по традиционной шкале
Студент освоил изучаемый материал (основной и дополнительный), системно и грамотно излагает его, осуществляет полное и правильное выполнение заданий в соответствии с индикаторами достижения компетенций, способен ответить на дополнительные вопросы.	75-100	<i>Отлично</i>
Студент освоил изучаемый материал, осуществляет выполнение заданий в соответствии с индикаторами достижения компетенций с не принципиальными ошибками.	50-74	<i>Хорошо</i>
Студент демонстрирует освоение только основного материала, при выполнении заданий в соответствии с индикаторами достижения компетенций допускает отдельные ошибки, не способен систематизировать материал и делать выводы.	25-49	<i>Удовлетворительно</i>
Студент не освоил основное содержание изучаемого материала, задания в соответствии с индикаторами достижения компетенций не выполнены или выполнены неверно.	<25	<i>Неудовлетворительно</i>

3. Типовые контрольные задания или иные материалы, необходимые для оценки уровня достижения компетенций в соответствии с индикаторами

1.Примеры вопросов по дисциплине

Компетенция	Индикатор достижения компетенции
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

ФОМ по дисциплине «Программирование кроссплатформенных систем»

1. Применяя знания в области разработки и отладки программы и их блоки, дайте понятие кроссплатформенного программного обеспечения.
2. Применяя знания в области разработки и отладки программы и их блоки, сформулируйте проблему переносимости приложений с одной ОС на другую.
3. Применяя знания в области разработки и отладки программы и их блоки, расскажите об использовании 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.
4. Применяя знания в области разработки и отладки программы и их блоки, дайте расскажите об использовании 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).
5. Применяя знания в области разработки и отладки программы и их блоки, расскажите о кроссплатформенных средах 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.
6. Применяя знания в области разработки и отладки программы и их блоки, расскажите о процессе установке и работе в Pydroid 3 на базе Android.
7. Применяя знания в области разработки и отладки программы и их блоки, расскажите 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.
8. Применяя знания в области разработки и отладки программы и их блоки, расскажите 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.
9. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику встроенным числовым типам данных. Арифметические операции с целыми и вещественными числами. Работа с комплексными числами.
10. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику битовым операциям 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.
11. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику конструкции 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.

12. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику кортежу (`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.
13. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику понятию 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.
14. Применяя знания в области разработки и отладки программы и их блоки, дайте характеристику инкапсуляции, наследованию и полиморфизму. 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.
15. Применяя знания в области разработки и отладки программы и их блоки, приведите пример работы со строками в Python.
16. Применяя знания в области разработки и отладки программы и их блоки, приведите примеры работы условных операторов и циклов. Работа со списками. 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.

17. Применяя знания в области разработки и отладки программы и их блоки, приведите пример работы с классами и объектами
18. Применяя знания в области разработки и отладки программы и их блоки, приведите пример работы с базой данных SQLite. Основы использования библиотеки PyQt5.

4. Файл и/или БТЗ с полным комплектом оценочных материалов прилагается.