

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Ястребов Олег Александрович
Должность: Ректор
Дата подписания: 27.02.2025 15:40:33
Уникальный программный ключ:
ca953a0120d891083f939673078ef1a989dae18a

Приложение к рабочей программе
дисциплины (практики)

**Федеральное государственное автономное образовательное учреждение
высшего образования «Российский университет дружбы народов имени
Патриса Лумумбы» (РУДН)**

Факультет искусственного интеллекта
(наименование основного учебного подразделения)

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ И СИСТЕМА ОЦЕНИВАНИЯ УРОВНЯ
СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ ПО ДИСЦИПЛИНЕ
(ПРАКТИКЕ)**

ЯЗЫКИ ПРОГРАМИРОВАНИЯ
(наименование дисциплины (практики))

**Оценочные материалы рекомендованы МССН для направления подготовки/
специальности:**

10.03.01 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ
(код и наименование направления подготовки/ специальности)

**Освоение дисциплины (практики) ведется в рамках реализации основной
профессиональной образовательной программы (ОП ВО, профиль/
специализация):**

**ОРГАНИЗАЦИЯ И ТЕХНОЛОГИИ ЗАЩИТЫ ИНФОРМАЦИИ (ПО ОТРАСЛИ
ИЛИ В СФЕРЕ ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ)**
(направленность (профиль) ОП ВО)

1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ ПО ДИСЦИПЛИНЕ (ПРАКТИКЕ)

Таблица 1. Шкала и критерии оценивания отчета по лабораторной работе

Шкала	Критерии оценивания
Оценка «зачтено» (начисляются все баллы, запланированные по конкретной лабораторной работе БРС)	<ul style="list-style-type: none"> - изложение материала логично, грамотно; - свободное владение терминологией; - умение высказывать и обосновать свои суждения при ответе на контрольные вопросы; - умение описывать изучаемые явления и процессы; - умение проводить и оценивать результаты измерений; - способность разрешать конкретные ситуации (допускается наличие малозначительных ошибок или недостаточно полное раскрытие содержания вопроса или погрешность непринципиального характера в ответе на вопросы).
Оценка «не зачтено» (баллы не начисляются)	<ul style="list-style-type: none"> - отсутствие необходимых теоретических знаний; допущены ошибки в определении понятий и описании изучаемых явлений и процессов, искажен их смысл, не правильно оцениваются результаты измерений; - незнание основного материала учебной программы, допускаются грубые ошибки в изложении.

Примерные оценочные средства для текущего контроля успеваемости

При текущем контроле успеваемости обучающийся должен предъявить во время практических занятий решенные задачи. Задача считается решенной после предъявления преподавателю программного кода на указанном языке, ее работы в тестовых условиях и объяснения алгоритма работы и подготовленного программного кода.

Примеры задач:

1. Написать программу шифрования текстовой строки шифром Цезаря. Строку для шифрования и ключ ввести с клавиатуры
1. Написать программу расшифровки текстовой строки, зашифрованной шифром Цезаря. Строку для расшифровки и ключ ввести с клавиатуры
2. Написать программу шифрования текстовой строки шифром Гронсфельда. Строку для шифрования и ключ ввести с клавиатуры
3. Написать программу расшифровки текстовой строки, зашифрованной шифром Гронсфельда. Строку для расшифровки и ключ ввести с клавиатуры
4. Написать программу шифрования текстовой строки аффинным шифром. Строку для шифрования и ключ ввести с клавиатуры
5. Написать программу расшифровки текстовой строки, зашифрованной аффинным шифром. Строку для расшифровки и ключ ввести с клавиатуры
6. Написать функцию `is_year_leap`, принимающую 1 аргумент — год, и возвращающую `True`, если год високосный, и `False` иначе.
7. Написать программу, выводящую все простые числа меньше заданного целого числа. Число ввести с клавиатуры.
8. Написать функцию `factorial`, вычисляющую факториал. Аргумент и возвращаемое число - целые.
9. Написать функцию быстрого возведения целого числа в целую степень

10. Написать функцию `factorial`, вычисляющую факториал методом разбиения на диапазоны (“дерево”). Аргумент и возвращаемое число - целые.
11. Написать программу, выводящую количество всех сочетаний из n по m элементов. Целые положительные m и n ввести с клавиатуры
12. Написать программу, подсчитывающую частоту букв русского алфавита в текстовом файле.
13. Написать программу удаления лишних пробелов в введенной с клавиатуры строке. Лишними считать дублирующиеся пробелы и пробелы перед знаками препинания
14. Написать программу составления словаря файла с текстом на русском языке. Вывести все слова без дублирования.

Примерные оценочные средства для контроля самостоятельной работы студентов.

Типовые контрольные вопросы и задания для проведения промежуточной аттестации:

1. Языки программирования - определение, классификации
2. Основные алгоритмические конструкции. Теорема Бёма-Якопини. Реализация основных конструкций и объявления данных в языке C.
3. Основные алгоритмические конструкции. Теорема Бёма-Якопини. Реализация основных конструкций и объявления данных в языке C++.
4. Основные алгоритмические конструкции. Теорема Бёма-Якопини. Реализация основных конструкций и объявления данных в языке Python.
5. Типы данных в языке C
6. Типы данных в языке C++
7. Типы данных в языке Python
8. Подпрограммы. Способы передачи параметров. Функции в языке C.
9. Подпрограммы. Способы передачи параметров. Функции в языке C++.
10. Подпрограммы. Способы передачи параметров. Функции в языке Python.
11. Структуры данных. Реализация одно- и двух-мерных массивов в языке C
12. Структуры данных. Списки и реализация односвязных списков на языке C
13. Структуры данных. Бинарные деревья и их реализация на языке C.
14. Строки и их обработка на языке C
15. Строки и их обработка на языке Python
16. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Принцип инкапсуляции. Классы в C++
17. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Принцип наследования. Классы в C++
18. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Принцип полиморфизма. Классы в C++
19. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Реализация классов в языке C++, секции классов.
20. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Реализация классов в языке C++, полиморфизм и перегрузка операторов.

21. Метапрограммирование и шаблоны в языке C++. Итераторы в C++
22. Функции и итераторы в Python. Генераторы, функции с сохранением состояния. Модули и импорт модулей.
23. Основы функционального программирования. Лямбда-функции на языке Python. Использование функций map, reduce, filter, zip в Python
24. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Принцип инкапсуляции. Классы в Python
25. Парадигма объектно-ориентированного программирования. Основные принципы объектно-ориентированного программирования. Принцип наследования. Классы в Python