ПРИЛОЖЕНИЕ А
ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ «Высокоуровневые методы информатики и программирования»

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

Код контролируемой компетенции	Способ оценивания	Оценочное средство
ПК-2: Способность разрабатывать и адаптировать прикладное программное обеспечение	Зачет	Комплект контролирующих материалов для зачета
ПК-3: Способность проектировать ИС по видам обеспечения	Зачет	Комплект контролирующих материалов для зачета

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 «Высокоуровневые методы информатики и программирования».

При оценивании сформированности компетенций по дисциплине «Высокоуровневые методы информатики и программирования» используется 100-балльная шкала.

Критерий	Оценка по 100-балльной шкале	Оценка по традиционной шкале
Студент освоил изучаемый материал, выполняет задания в соответствии с индикаторами достижения компетенций, может допускать отдельные ошибки.	25-100	*Зачтено*
Студент не освоил основное содержание изученного материала, задания в соответствии с индикаторами достижения компетенций не выполнены или выполнены неверно.	0-24	*Не зачтено*

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

1.Разработать алгоритм для решения задачи. Используя инструментальные средства, разработать программу и спроектировать интерфейс для реализации поставленной задачи.

Компетенция	Индикатор достижения компетенции
ПК-2 Способность разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1 Разрабатывает алгоритм решения задачи
	ПК-2.2 Создает программный код на языке

	программирования
	ПК-2.3 Применяет инструментальные средства разработки и адаптации прикладного программного обеспечения
ПК-3 Способность проектировать ИС по видам обеспечения	ПК-3.3 Выполняет проектирование структур данных и интерфейсов по предъявленным требованиям к ИС

Дисциплина Высокоуровневые методы информатики и программирования

Факультет информационных технологий
Кафедра «Информационные системы в экономике»
Направление 09.03.03 Прикладная информатика
Профиль Прикладная информатика в экономике

Компетенции и индикаторы их достижения

Компетенция	Содержимое компетенции	Индикатор	Содержимое индикатора
ПК-2	Способность разрабатывать и адаптировать прикладное программное обеспечение	ПК-2.1	Разрабатывает алгоритм решения задачи
		ПК-2.2	Создает программный код на языке программирования
		ПК-2.3	Применяет инструментальные средства разработки и адаптации прикладного программного обеспечения
ПК-3	Способность проектировать ИС по видам обеспечения	ПК-3.3	Выполняет проектирование структур данных и интерфейсов по предъявленным требованиям к ИС

Варианты заданий:

1. Дан файл со словами (по 1 слову в строке). Найти такую пару слов, что суффикс одного является префиксом другого, причём суффикс имеет наибольшую длину

2. Реализовать генератор простых чисел-близнецов

3. Реализовать функцию поиска элемента в бинарном дереве

4. Даны функция f() и числа a, b. Известно, что $f(a) < 0$, $f(b) > 0$. Найти решение: $f(x) = 0$

5. Даны две строки. Проверить, является ли одна из них циклическим сдвигом другой

6. Дана строка. Построить все уникальные подстроки и вывести с сохранением порядка следования

7. Дана строка. Построить частотный словарь символов и вывести по убыванию частот

8. Реализовать класс дробей с операциями + - * /

9. Есть 100 дверей, изначально закрытых. Делается 100 проходов. На первом проходе посещаем каждую дверь (посетить = сменить состояние на противоположное), на втором - каждую вторую дверь(0, 2, 4, 6) , на третьем каждую третью. Найти конечное состояние дверей.

10. Анаграмма - слово полученное перестановкой букв другого слова, например: апельсин — спаниель, покраснение — пенсионерка. Написать функцию, is_anagram(a, b), возвращающую True если a и b анаграммы. Более сложный вариант: дан файл со словами, найти в нём все анаграммы.

11. Дан произвольный массив, найти моду (значение которое встречается чаще всего). Если мод несколько, вернуть None.

Как вариант - вернуть массив из мод.

12. Написать класс, способный вычислять простое скользящее среднее с размером окна 100 (среднее арифметическое последних 100 значений)

Методы: add_value, get_moving_average.

13. Дана строка. Определить сбалансированы ли квадратные скобки в ней. Пример хороших строк: "", "[]", "[][]", "[[][]]". Пример плохих строк: "][", "][[", "[][]["

14. Дана строка. Случайно перемешать в ней символы, но так чтобы ни один символ не остался на своём месте.

15. Дан отсортированный массив. Найти в нём элемент, используя бинарный поиск.

16. Определить является ли число степенью двойки, не используя if.

17. Написать функцию для игры "быки и коровы" calc(entered_num, right_num), которая возвращает tuple из двух чисел - количество цифр в entered_num на своих местах, и количество цифр не на своих местах)

18. Дан массив из чисел. Найти в нём пару самых "близких" чисел (чисел с минимальным модулем разности)

19. Написать генератор, возвращающий все перестановки элементов в исходном массиве. Порядок не важен.

Пример: для [1, 2, 3] -> [1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]

20. Дано, число, разложить его на простым множители и вернуть их число.

21. Дано число, найти сумму его цифр.

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