

**ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

1. Общие сведения

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	01.03.02 Прикладная математика и информатика
3.	Направленность (профиль)	Системное программирование и компьютерные технологии
4.	Дисциплина (модуль)	Б1.О.15.05 Функциональное и логическое программирование
5.	Форма обучения	Очная
6.	Год набора	2023

2. Перечень компетенций

– ОПК-2: Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач
--

3. Критерии и показатели оценивания компетенций на различных этапах их формирования

Этап формирования компетенции (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
Понятие об искусственном интеллекте, интеллектуальных и экспертных системах	ОПК-2	понятие искусственный интеллект; назначение и области применения искусственного интеллекта; роль изучения искусственного интеллекта для науки и производства.	использовать основные приемы работы с готовыми базами знаний.	технологией анализа функционала интеллектуальных систем.	контрольные вопросы по теме; реферат.
Базовые принципы языков функционального и логического программирования	ОПК-2	основы типизации функций и данных; значение функции как описания процесса вычисления; некоторые модели и абстрактные типы данных; понятия: класс, подкласс, суперкласс, объект, полиморфизм, наследование; способы описания классов средствами ЯФП; стандартные классы (суперклассы) одного из ЯФП.	выполнять формальное определение типов данных; методы вызова функций, в т.ч. анонимных; применять операции композиции функций; определять место текущего класса в общей иерархии классов; задавать класс и определять на основании него объекты, подклассы; переопределять отдельные операторы классов; удалять объекты и классы.	технологией описания функциональных типов; технологией работы с классами и объектами на примере суперклассов, собственных классов.	выполнение и защита лабораторных работ; решение задач по теме; контрольные вопросы по теме.
Комбинаторная логика и лямбда-исчисление	ОПК-2	понятие и назначение лямбда-функции; основы комбинаторной логики и возможности ее применения в ФП; значение абстракции функций для вычислительных процессов.	определять и вызывать собственные лямбда-функции; использовать инструментарий стандартных лямбда-функций.	технологией построения формальной системы как некоторого формального исчисления.	выполнение и защита лабораторных работ; решение задач по теме; контрольные вопросы по теме.
Основы логического программирования	ОПК-2	историю развития логического программирования; основные свойства ЯЛП; круг задач, решаемых методами ЯЛП.	определять степень адекватности ЯЛП для решения конкретной задачи; выполнять конструирование функций, в т.ч. рекурсивных; определять область видимости переменных функции соответственно решаемой задаче.	технологией доказательства свойств функций; методологией написания программ средствами ЯЛП в интегрированных средах разработки.	выполнение и защита лабораторных работ; решение задач по теме; контрольные вопросы по теме.
Основы функционального программирования	ОПК-2	историю развития функционального программирования; основные свойства ЯФП; круг задач, решаемых методами ЯФП.	определять степень адекватности ЯФП для решения конкретной задачи; выполнять конструирование функций, в т.ч. рекурсивных; определять область видимости переменных функции соответственно решаемой задаче.	технологией доказательства свойств функций; методологией написания программ средствами ЯФП в интегрированных средах разработки.	выполнение и защита лабораторных работ; решение задач по теме; контрольные вопросы по теме.

Шкала оценивания в рамках балльно-рейтинговой системы:

«неудовлетворительно» – 60 баллов и менее; «удовлетворительно» – 61-80 баллов; «хорошо» – 81-90 баллов; «отлично» – 91-100 баллов

4. Критерии и шкалы оценивания

4.1 Входное тестирование

Процент правильных ответов	До 40	41-60	61-80	81-90	91-100
Количество баллов за решенный тест	0	1	2	3	4

4.2 Тест по разделу дисциплины (Защита модуля)

Процент правильных ответов	До 60	61-80	81-100
Количество баллов за решенный тест	0	1	2

4.3 Критерии оценки выполнения лабораторной работы

Содержание отчета	Балл
Все упражнения и задания лабораторной работы выполнены полностью и своевременно, все материалы оформлены в соответствии с требованиями	3
Данная оценка выставляется в следующих случаях: <ul style="list-style-type: none">– выполнено не менее 60% упражнений и заданий лабораторной работы– требования к оформлению материалов соблюдены частично– работа выполнена полностью, но представлена после установленных сроков сдачи	2
Данная оценка выставляется в следующих случаях: <ul style="list-style-type: none">– выполнено менее 60% упражнений и заданий лабораторной работы– требования к оформлению материалов соблюдены частично– работа выполнена полностью, но представлена существенно позже установленных сроков сдачи	1
Задания лабораторной работы не выполнены, выполнены неудовлетворительно либо невозможно установить авторство	0

4.4 Подготовка реферата, участие в учебной дискуссии:

Критерии оценивания содержательной части реферата	0-3 балла
Выполнены все требования к содержательной и оформительской части доклада: <ul style="list-style-type: none">– текст доклада соответствует теме, тема раскрыта достаточно полно, сделаны необходимые выводы и обобщения, теоретические сведения проиллюстрированы примерами– доклад оформлен в соответствии с требованиями к оформлению– при подготовке доклада использовано не менее трех источников	3
При оформлении текста доклада допущены недочеты, не влияющие на его содержательную часть	2
Оценка выставляется, если: <ul style="list-style-type: none">– тема доклада раскрыта слабо или неполно– в тексте отсутствуют выводы, обобщения, приведены частные примеры– оформление текста не соответствует требованиям	1
Оценка выставляется, если: <ul style="list-style-type: none">– текст доклада не представлен– тема доклада не раскрыта, либо из текста можно сделать вывод о том, что студент не разобрался в материале– текст в значительной мере заимствован из одного или нескольких источников– оформление текста не соответствует требованиям	0
Критерии оценивания выступления	0-2 балла
Выполнены все требования к публичной защите доклада: <ul style="list-style-type: none">– во время выступления использованы наглядные материалы (презентация, иллюстрации, схемы)– ответы на уточняющие вопросы демонстрируют понимание студентом темы, аргументированы и подкреплены как теоретическими сведениями, так и практическими примерами	2
Ответы на вопросы неполны либо отсутствуют	1
Выступления нет либо оно проведено неудовлетворительно	0

4.5 Написание эссе:

Критерии оценивания	Балл
Содержание эссе	0-3
Выполнены все требования к эссе: <ul style="list-style-type: none">– студент глубоко и всесторонне усвоил теоретический материал, уверенно, логично, последовательно и грамотно его излагает– опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные знания с практической деятельностью	3

<ul style="list-style-type: none"> – делает выводы и обобщения – свободно владеет понятиями – способен описать круг функциональных задач, решаемых на базе имеющихся знаний по разделу 	
Ответ студента в целом верен и достаточно полный, однако содержит неточности и недочеты, не позволяющие выставить 2 балла	2
Ответ демонстрирует наличие у студента некоторых знаний по теме эссе	1
Ответ отсутствует или неудовлетворителен	0
Количественные показатели	0-2
<ul style="list-style-type: none"> – количество слов не менее 500 и не более 800 – соответствие содержания заявленной тематике – наличие ссылок на источники – оригинальность работы – не менее 61% 	По 0,5 баллов за каждый пункт

4.6 Контрольное (экзаменационное) тестирование: балл рассчитывается пропорционально количеству верно решенных дидактически единиц (модулей):

Количество верно решенных ДЕ	0-5
Количество баллов	По 8 баллов за каждую ДЕ

5. Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

1. Функциональное программирование относится к:

- а) императивному программированию;
- б) объектно-ориентированному программированию;
- в) декларативному программированию.

2. Декларативный подход в программировании требует:

- а) описывать последовательность действий для нахождения результата;
- б) описывать свойства требуемого результата;
- в) описывать последовательность действий и свойства требуемого результата.

3. К особенностям декларативных языков программирования относятся:

- а) выразительность, параллелизм и ориентированность на присущий человеку образ мышления;
- б) параллелизм, полиморфизм и наследование;
- в) выразительность, инкапсуляция и ориентированность на присущий человеку образ мышления.

4. К языкам функционального программирования относятся:

- а) Lisp, Scheme, Miranda, Haskell, O’Caml, F#, Erlang.
- б) Lisp, Haskell, Prolog, O’Caml, F#, Clean.
- в) Lisp, Haskell, F#, C#, C++, Object Pascal.
- г) Lisp, Miranda, Haskell, JavaScript, Visual Prolog, O’Caml, F#, Erlang.

5. К языкам логического программирования относятся:

- а) Haskell, Cofer.
- б) Prolog, Visual Prolog.
- в) Prolog, F#.

6. Язык функционального программирования F# наследует от языков:

- а) Lisp, O’Caml.
- б) Lisp, Haskell.
- в) O’Caml, Haskell.

7. Тип выражения говорит о том:

- а) какие значения могут получиться при вычислении этого выражения.
- б) имеет ли выражение какое-нибудь значение или нет.
- в) какие значения могут получиться при вычислении этого выражения и получатся ли какие-нибудь значения вообще.

8. Базовый элемент синтаксиса F#:

- а) константа;
- б) выражение;
- в) объект;
- г) класс.

9. Функции в F# объявляются конструкцией:

- а) <имя функции> (<аргумент функции>) : = <тело функции>;
- б) let <имя функции> (<аргумент функции>) = <тело функции>;
- в) let <тип функции> <имя функции> (<аргумент функции>) : {<тело функции>;};

10. Функция let plus x y = x+y;; будет иметь тип:

- а) int -> int;
- б) int -> int -> int;
- в) int*int -> int.

11. Результат выражения 0.1+1 будет:

- а) 1.1;
- б) 1;
- в) 0.1;
- г) ошибка.

12. Каррирование – это прием функционального программирования, который:

- а) функцию с более чем одним аргументом интерпретирует как функцию от первого аргумента, возвращающую функцию от оставшихся, при фиксированном значении первого.
- б) позволяет выражениям и функциям иметь более одного типа.
- в) функцию с более чем одним аргументом интерпретирует как функцию от первого аргумента при его фиксированном значении.

12. Аппликативный порядок применения функции- это порядок, при котором:

- а) аргументы функции начинают вычисляться только тогда, когда в них возникает необходимость;
- б) аргументы функции полностью вычисляются до ее вызова.

13. Лямбда-выражение $\lambda x.x^2-x$ в F# будет записано в виде:

- а) let f x = x*x-x;
- б) fun x -> x*x-x;
- в) let function x -> x*x-x.

14. Задана функция let sum x = x+x;; При нормальном порядке применения функции sum 3+2 следующий шаг будет:

- а) sum 5;
- б) (3+2) + (3+2);
- в) 5+5;
- г) 10.

15. Список – это:

- а) конечная последовательность элементов одного типа;
- б) бесконечная последовательность элементов одного типа;
- в) конечная последовательность элементов разных типов.

16. Хвостом списка называется:

- а) первый элемент списка;
- б) последний элемент списка;
- в) все элементы списка кроме первого.

17. Результат сопоставления `let x::y::z = [1;2;3;4]` будет:

- а) `x=[1;2], y=3, z=4;`
- б) `x=1, y=2, z=[3;4];`
- в) `x=1, y=2, z=[];`
- г) `x=1, y=2, z=3.`

18. Функция конкатенации («склейки») списков `[1;2;3]` и `[4;5]` будет записана как:

- а) `[1;2;3] + [4;5];`
- б) `[1;2;3][4;5];`
- в) `[1;2;3] and [4;5];`
- г) `[1;2;3] @ [4;5].`

19. Сложность добавления элемента в начало списка длины `n` будет составлять:

- а) $O(1)$;
- б) $O(n)$;
- в) $O(n^2)$;
- г) $O(1+n)$.

20. Извлечь из массива `A`, состоящего из первых `6` элементов, подмассив с 3-го по 5-й элемент можно в виде:

- а) `A.[3..5];`
- б) `A[2,3,4];`
- в) `A.[2..4];`
- г) `[A[3]; A[4]; A[5]].`

21. В классических функциональных языках многомерные массивы представляются:

- а) списками массивов;
- б) списками списков;
- в) массивами списков.

22. Подмассиву с 3-го по 5-й элемент массива `A` присвоить новые значения больше их индекса на 1 можно в виде:

- а) `A.[2..4] <- [3..5];`
- б) `A.[3..5] = [4..6];`
- в) `A[2..4] -> 3..5;`
- г) `A[3..5] = [4; 5; 6].`

23. Матрицы в библиотеке `F#` имеют тип:

- а) `Matrix[][]`;
- б) `Vector<_>`;
- в) `Matric()`;
- г) `Matrix<_>`.

Ключ к тестовым заданиям

№ вопроса	1	2	3	4	5	6	7	8	9	10	11	12
Ответ	В	Б	А	А	Б	В	В	Б	Б	Б	Г	А
№ вопроса	13	14	15	16	17	18	19	20	21	22	23	-
Ответ	Б	Б	А	В	Б	Г	А	В	Б	А	Г	-

Примерные вопросы к зачету

1. Парадигма функционального программирования
2. Основные свойства функциональных языков
3. Типовые задачи, решаемые методами функционального программирования
4. Конструирование функций
5. Доказательство свойств функций. Примеры.
6. Списки. Операции со списками. Примеры.
7. Использование функций для описания процессов вычисления.
8. Типизация данных и функций.
9. Модули и абстрактные типы данных.

10. Понятие о полиморфизме. Область применения полиморфизма. Примеры.
11. Понятие о наследование. Примеры реализации.
12. Классы как способ абстракции. Реализация работы с классами на примере одного из ЯФП.
13. Стандартные классы (суперклассы) на примере одного из ЯФП.
14. Понятие монады. Монада как тип-контейнер.
15. Монады. Последовательное выполнение действий.
16. Стандартные монады на примере одного из ЯФП.
17. Разработка собственных монад.
18. Основы комбинаторной логики.
19. Лямбда-исчисление как теоретическая основа функционального программирования.
20. Редукция и вычисления в функциональных языках.
21. Понятие о математической лингвистике.
22. Понятие транслятора. Теория построения трансляторов средствами ЯФП.
23. Возможности библиотек для создания трансляторов.
24. Понятие об искусственном интеллекте. Основные задачи искусственного интеллекта.
25. Перспективы функционального программирования.