

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Институт информационных технологий, математики и механики

УТВЕРЖДЕНО
решением Ученого совета ННГУ
протокол № 10 от 02.12.2024 г.

Рабочая программа дисциплины
Современные проблемы дискретной математики

Уровень высшего образования
Магистратура

Направление подготовки / специальность
01.04.02 - Прикладная математика и информатика

Направленность образовательной программы
Компьютерные науки и приложения

Форма обучения
очная

г. Нижний Новгород

2025 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.В.03 Современные проблемы дискретной математики относится к части, формируемой участниками образовательных отношений образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ПК-11: Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых задач производственно-технологической деятельности	<p>ПК-11.1: Знает методы разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p> <p>ПК-11.2: Умеет применять методы разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p> <p>ПК-11.3: Имеет навыки применения методов разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач</p>	<p>ПК-11.1: Знать особенности языков функционального программирования. Знать основы языка программирования Racket Знать разные виды рекурсии, их преимущества и недостатки.</p> <p>ПК-11.2: Уметь писать программы на языке Racket с использованием рекурсии, функций высшего порядка и побочных эффектов. Уметь понимать сообщения об ошибках и исправлять ошибки в программе.</p> <p>ПК-11.3: Владеть навыками отладки программ. Владеть навыками составления эффективных рекурсивных алгоритмов. Владеть навыками чтения и понимания большого объема кода на языке Racket.</p>	Контрольная работа	Экзамен: Контрольные вопросы Задачи
ПК-4: Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых	ПК-4.1: Знает методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем	<p>ПК-4.1: Знать основы лямбда-исчисления. Знать семантику ядра языка Racket.</p>	Доклад-презентация	Экзамен: Проект Контрольные вопросы

научных проблем и задач	и задач ПК-4.2: Умеет применять методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач ПК-4.3: Имеет навыки применения методов разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач	Знать различия между энергичной и ленивой семантиками. ПК-4.2: Уметь реализовывать интерпретатор ядра языка Racket. Уметь добавлять новые конструкции к интерпретатору языка программирования. Уметь объяснять поведение программы с помощью диаграмм окружений. ПК-4.3: Владеть навыками составления доказательств свойств лямбда-исчисления.		
-------------------------	--	--	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	4
Часов по учебному плану	144
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	32
- КСР	2
самостоятельная работа	42
Промежуточная аттестация	36 Экзамен

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			Самостоятельная работа обучающегося, часы
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			
		Занятия лекционного типа	Занятия семинарского типа (практические	Всего	

			занятия/лабораторные работы), часы		
	о Ф о	о Ф о	о Ф о	о Ф о	о Ф о
1. Основы программирования на языке Racket.	40	12	12	24	16
2. Основы лямбда-исчисления.	20	6	6	12	8
3. Метациклический интерпретатор.	20	6	6	12	8
4. Дополнительные возможности языков программирования	26	8	8	16	10
Аттестация	36				
КСР	2			2	
Итого	144	32	32	66	42

Содержание разделов и тем дисциплины

1. Введение в Racket. Рекурсия и итерация. Функции высших порядков. Пары и списки. Сравнение. Функции высших порядков на списках. Локальные определения. Побочные эффекты.
2. Введение в лямбда-исчисление. Свойство Чёрча-Россера. Разрешимость бета-равенства для сильно нормализуемых термов. Модель вычислений с окружениями.
3. Семантика Racket. Диаграммы окружений. Структура интерпретатора Racket. Добавление новых особых форм в ядро интерпретатора и в качестве синтаксического расширения.
4. Ленивые вычисления. Потoki. Объектно-ориентированное программирование. Параллелизм. Недетерминизм. Логическое программирование.

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

Абельсон Х., Сассман Дж. Структура и интерпретация компьютерных программ. Добросвет, 2006.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Контрольная работа) для оценки сформированности компетенции ПК-11:

1. Дано следующее определение: `(define p (lambda () (p)))`. Что произойдет при вычислении `p`, `(p)`, `((p))`? Если вычисление возвращает результат, напишите его, а также его тип.
2. Рассмотрим следующее определение: `(define lst '(() (((1 a) 2) 3 4) 5))`. Напишите выражение, содержащее только `car`, `cdr` и `lst` и возвращающее символ `a`.

3. Дано следующее определение.

```
(define (iter lst answer)
  (if (null? lst)
      answer
      (iter (cdr lst)
            (cons answer (add1 (car lst))))))
```

Напишите трассировку вычисления выражения (iter '(1 2 3) '(4)).

Критерии оценивания (оценочное средство - Контрольная работа)

Оценка	Критерии оценивания
зачтено	Все задачи решены полностью или решение задач обосновано, но допущены арифметические ошибки. Задачи не решены или сделан только первый этап в решении задач.
не зачтено	Задачи не решены или сделан только первый этап в решении задач.

5.1.2 Типовые задания (оценочное средство - Доклад-презентация) для оценки сформированности компетенции ПК-4:

Темы докладов.

1. Ленивые вычисления.
2. Потоки.
3. Объектно-ориентированное программирование. Множественные представления для абстрактных данных.
4. Параллелизм.
5. Недетерминизм.
6. Логическое программирование.

Критерии оценивания (оценочное средство - Доклад-презентация)

Оценка	Критерии оценивания
зачтено	Доклад охватывает весь заданный материал. Доклад была понятна слушателям. По окончании доклада слушатели могут ответить на вопросы по теме презентации. Использовались лучшие практики по составлению презентации.
не зачтено	Доклад не включает существенную часть материала. Слушателям было сложно понимать доклад. По окончании доклада слушатели не могут ответить на вопросы по теме презентации. Сложно читать слайды, присутствуют ненужные и отвлекающие визуальные эффекты.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые с недочетами	Продемонстрированы все основные умения. Решены все основные задачи с отдельными несущественными недочетами, выполнены все задания в полном объеме	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-11

1. Использование нехвостостовой рекурсии и хвостовой рекурсии (итерации) в языках функционального программирования. Их преимущества и недостатки. Примеры.
2. Пары и списки. Списочные диаграммы.
3. Функции высших порядков на числах и на списках. Примеры: map, filter, левая и правая свертки.
4. Основная теорема о рекуррентных соотношениях. Определение временной и пространственной сложности функций.
5. Локальные переменные. Формы let и letrec.
6. Присваивание. Отличия программирования с побочными эффектами от чистого функционального программирования.

5.3.2 Типовые задания (оценочное средство - Контрольные вопросы) для оценки сформированности компетенции ПК-4

1. Лямбда-исчисление: определение термов и β -редукции. Примеры вычисления лямбда-термов.
2. Модель вычисления с окружениями. Необходимость замыкания функций.
3. Семантика с окружениями ядра Racket. Диаграммы окружений.
4. Вычисление интерпретатора в нем самом.
5. Нормальный и аппликативный порядок вычислений. Интерпретатор с ленивым вычислением.
6. Потоки и задержанное вычисление.
7. Параллелизм. Его реализация с помощью сериализаторов, мьютексов и семафоров.

Критерии оценивания (оценочное средство - Контрольные вопросы)

Оценка	Критерии оценивания
превосходно	свободное владение основным и дополнительным материалом без ошибок и погрешностей.
отлично	свободное владение основным материалом с незначительными ошибками и погрешностями.
очень хорошо	достаточное владение основным материалом с незначительными погрешностями.
хорошо	владение основным материалом с рядом заметных погрешностей.
удовлетворительно	владение минимальным материалом, необходимым по данному предмету, с рядом ошибок.
неудовлетворительно	владение материалом недостаточно, необходима дополнительная подготовка.
плохо	отсутствие владения материалом.

5.3.3 Типовые задания (оценочное средство - Задачи) для оценки сформированности компетенции ПК-11

1. С помощью `foldl` напишите функцию (`horner poly x`), которая возвращает значение на аргументе `x` многочлена `poly`, заданного списком своих коэффициентов. Старший коэффициент находится в голове списка.

```
> (horner '(1 0 -2 3) 2)
```

```
7
```

2. Пусть матрица задана списком своих строк. Без использования рекурсии, но с помощью функций высших порядков напишите функцию (`matrix-*-matrix m n`), которая возвращает произведение матриц `m` и `n`.

3. Напишите функцию `nested-pairs`, поведение которой иллюстрируется следующим примером.

```
> (nested-pairs 7)
```

```
(1 (2 (3 (4 (5 (6 (7)))))))
```

Критерии оценивания (оценочное средство - Задачи)

Оценка	Критерии оценивания
превосходно	Все задачи решены в полном объеме без недочетов.
отлично	Все задачи решены с отдельными несущественными недочетами.
очень хорошо	Все основные задачи решены, но некоторые с недочетами.

Оценка	Критерии оценивания
хорошо	Все основные задачи решены с негрубыми ошибками.
удовлетворительно	Типовые задачи решены с негрубыми ошибками.
неудовлетворительно	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки.
плохо	Отсутствие минимальных умений решения задач. Невозможность оценить наличие умений.

5.3.4 Типовые задания (оценочное средство - Проект) для оценки сформированности компетенции ПК-4

1. Выражение (`while test expr1 ... exprn`) проверяет, истинно ли значение `test`. Если оно ложно (то есть равно `#f`), то вычисление заканчивается с значением `#void`. Если значение `test` истинно (отлично от `#f`), последовательно вычисляются `expr1, ..., exprn`, после чего снова проверяется `test` и т.д.

Напишите функцию `transform-while`, переводящую форму `while` в эквивалентное выражение без `while`. Добавьте форму `while` к интерпретатору в качестве синтаксического расширения.

2. Выражение (`until expr1 ... exprn test`) последовательно вычисляются `expr1, ..., exprn`, после чего вычисляет значение `test`. Если оно ложно (то есть равно `#f`), то снова вычисляются значения `expr1, ..., exprn` и т.д. Если значение `test` истинно (отлично от `#f`), вычисление заканчивается с неопределенным значением `#void`.

Напишите функцию `transform-until`, переводящую особую форму `until` в эквивалентное выражение без `until`. Добавьте форму `until` к интерпретатору в качестве синтаксического расширения.

3. Добавьте в ядро интерпретатора особую форму `for` с несколькими переменными. Например,

```
(for ([i '(1 2 3)] [j '(a b c)]) (displayln (list i j)))
```

печатает

(1 a)

(2 b)

(3 c)

Во время цикла переменные параллельно продвигаются каждая по своему списку. Цикл заканчивается, когда заканчивается самый короткий список. Значение каждого выражения в теле цикла игнорируется, и значением всего выражения является `#<void>`.

Критерии оценивания (оценочное средство - Проект)

Оценка	Критерии оценивания
превосходно	Программа компилируется без ошибок и работает правильно. Используемый алгоритм является эффективным. Исходный код правильно оформлен, снабжен комментариями и использует лучшие практики программирования.
отлично	Программа компилируется без ошибок и работает правильно. Используемый алгоритм является эффективным. Оформление кода и комментарии содержат незначительные погрешности.
очень хорошо	Программа компилируется без ошибок и работает правильно. Используемый алгоритм является эффективным. Оформление кода содержит незначительные погрешности. Используемый алгоритм содержит отдельные неэффективности. Недостаточно комментариев.
хорошо	Программа компилируется без ошибок и работает правильно. Оформление кода содержит погрешности. Эффективность алгоритма можно улучшить. Недостаточно комментариев.
удовлетворительно	Программа компилируется без ошибок, но не во всем работает правильно. Оформление кода непоследовательное. Эффективность алгоритма можно улучшить. Комментарии отсутствуют.
неудовлетворительно	Программа не компилируется или работает неправильно. Алгоритм неэффективный. Оформление кода непоследовательное.
плохо	Программа не компилируется или не была передана преподавателю.

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Кубенский А. А. Функциональное программирование / Кубенский А. А. - Москва : Юрайт, 2022. - 348 с. - (Высшее образование). - URL: <https://urait.ru/bcode/490015> (дата обращения: 05.01.2022). - ISBN 978-5-9916-9242-7 : 1079.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=786689&idb=0>.
2. Зыков С. В. Программирование. Функциональный подход : учебник и практикум / С. В. Зыков. - 2-е изд. - Москва : Юрайт, 2023. - 150 с. - (Высшее образование). - ISBN 978-5-534-16942-3. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=871965&idb=0>.

Дополнительная литература:

1. Довек Ж. Введение в теорию языков программирования : учебное пособие / Довек Ж.; Леви Ж.-Ж. - Москва : ДМК-пресс, 2015. - 134 с. - ISBN 978-5-97060-242-3.,

<https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=868693&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

Сайт курса 6.001 в MIT: <https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005>.

Сайт курса 6.037 в MIT: <https://web.mit.edu/6.001/6.037>.

Язык Racket: <https://racket-lang.org>.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду.

Программа составлена в соответствии с требованиями ОС ННГУ по направлению подготовки/специальности 01.04.02 - Прикладная математика и информатика.

Автор(ы): Макаров Евгений Маратович.

Заведующий кафедрой: Золотых Николай Юрьевич, доктор физико-математических наук.

Программа одобрена на заседании методической комиссии от 02.12.2024, протокол № 5.