

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  
президиумом Ученого совета ННГУ  
протокол от  
«30» ноября 2022 г. № 13

Рабочая программа дисциплины

**Современные проблемы дискретной математики**

---

Уровень высшего образования

**магистратура**

---

Направление подготовки

**010402 Прикладная математика и информатика**

---

Направленность образовательной программы

**Компьютерные науки и приложения**

---

Форма обучения

**очная**

---

Нижегород  
2022

# 1. Место и цели дисциплины в структуре ООП

## Место дисциплины в структуре ООП

Данная дисциплина (Б1.В.06) читается в первом семестре магистратуры, относится к части, формируемой участниками образовательных отношений. Дисциплина опирается на материал курсов «Дискретная математика», «Математическая логика», «Языки и методы программирования», «Алгоритмы и структуры данных».

## Цели освоения дисциплины

Целями дисциплины являются знакомство с различными парадигмами программирования (императивным, функциональным, логическим и объектно-ориентированным), энергичными и ленивыми структурами данных, методами интерпретацией и компиляцией программ.

## 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	
ПК-4 Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых научных проблем и задач	ПК-4.1. Знает методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач	ЗНАТЬ основы языка Racket основы лямбда-исчисления понятия абстракции с помощью процедур и с помощью данных	Собеседование (экзамен)
	ПК-4.2. Умеет применять методы разработки и анализа концептуальных и теоретических моделей решаемых научных проблем и задач	УМЕТЬ писать программы на языке Racket с использованием рекурсии, функций высших порядков и побочных эффектов использовать процедурные абстракции и абстракции данных для структурирования программ	Контрольная работа (текущий контроль)
ПК-11 Способен разрабатывать и анализировать концептуальные и теоретические модели решаемых задач производственно-технологической деятельности	ПК-11.1. Знает методы разработки и анализа концептуальных и теоретических моделей решаемых производственно-технологических задач	ЗНАТЬ различные виды семантик языков программирования принципы интерпретации и компиляции языка функционального программирования	Собеседование (экзамен)
	ПК-11.2. Умеет применять методы разработки и анализа концептуальных и теоретических моде-	УМЕТЬ реализовывать различные семантики языков программирования реализовывать интерпретатор и компилятор языка функционального программирования	Программный проект

	лей решаемых производственно-технологических задач		
--	--	--	--

### 3. Структура и содержание дисциплины

#### 3.1. Трудоемкость дисциплины

	Очная форма обучения
Общая трудоемкость	4 ЗЕТ
Часов по учебному плану	144
в том числе	
аудиторные занятия (контактная работа):	66
- занятия лекционного типа	32
- занятия семинарского типа	32
- занятия лабораторного типа	
- текущий контроль (КСР)	2
самостоятельная работа	42
Промежуточная аттестация – экзамен	36

#### 3.2. Содержание дисциплины

Наименование и краткое содержание разделов и тем дисциплины	Всего (часы)	В том числе				
		Контактная работа (работа во взаимодействии с преподавателем), часы.				Самостоятельная работа обучающегося, часы
		Из них				
		Занятия лекционного типа	Занятия семинарского типа	Занятия лабораторного типа	Всего	
1. Введение в Racket. Рекурсия и итерация	7	2	2		4	3
2. Функции высших порядков. Пары и списки	7	2	2		4	3
3. Сравнение. Функции высших порядков на списках	7	2	2		4	3
4. Лямбда-исчисление. Побочные эффекты	7	2	2		4	3
5. Модель вычислений с окружениями	7	2	2		4	3
6. Метациклический интерпретатор	11	4	4		8	3
7. Множественные представления абстрактных данных	7	2	2		4	3
8. Параллелизм	7	2	2		4	3
9. Потoki	7	2	2		4	3
10. Ленивый интерпретатор	7	2	2		4	3
11. Недетерминированное вычисление	7	2	2		4	3

12. Логическое программирование	7	2	2		4	3
13. Регистровые машины	7	2	2		4	3
14. Компиляция	11	4	4		8	3
Текущий контроль (КСР)	2				2	
Промежуточная аттестация – экзамен	36					
Итого	144	32	32		66	42

Текущий контроль успеваемости реализуется в формах опросов на занятиях семинарского типа. Промежуточная аттестация проходит в традиционной форме (экзамен).

## 4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Ниже приводятся виды самостоятельной работы студентов, порядок их выполнения и контроля, приводится учебно-методическое обеспечение самостоятельной работы по ее отдельным видам и разделам дисциплины.

### Виды самостоятельной работы студентов:

- проработка теоретического материала лекционных занятий;
- подготовка домашних заданий к научно-практическим занятиям;
- подготовка к выполнению письменных контрольных работ;
- выполнение программного проекта;
- подготовка к промежуточной аттестации в форме экзамена.

### 4.1. Проработка теоретического материала лекционных занятий

Выполняется самостоятельно с использованием лекционных материалов и материалов, разобранных в литературе (список обязательной и дополнительной литературы приводится).

Контроль выполняется в форме проведения ежемесячного письменного экспресс-опроса по понятиям, фактам, формулировкам, выполняемого в течение 15 минут на практических занятиях. Экспресс – опрос оценивается оценками «Зачтено» – «Не зачтено».

### 4.2. Подготовка домашних заданий к научно-практическим занятиям

Домашние задания выдаются на практических занятиях.

Проверка выполнения домашних заданий проводится в начале каждого практического занятия. Используется две формы контроля: – выборочная проверка выполнения заданий у двух-трех человек из группы; – проверка в форме коллективного обсуждения у доски результатов выполнения отдельных заданий одним или двумя студентами.

### 4.3. Подготовка к выполнению письменных контрольных работ

В течение учебного семестра проводится одна контрольная работа по материалам лекций первой части курса.

Для подготовки к контрольным работам рекомендуется повторно прочитать теоретические разделы в задачнике, просмотреть полезные разделы в соответствующих источниках из списка рекомендованной литературы, а также самостоятельно решать несколько задач по теме контрольной работы в этих источниках.

### 4.4. Подготовка к выполнению программного проекта

В конце первой половины курса студентам предлагаются индивидуальные задания по расширению интерпретатора фрагмента языка Racket.

Для подготовки к выполнению этого проекта нужно изучить и понять выданный студентам текст интерпретатора, а также изучить решения нескольких аналогичных задач, выполненных на практиках.

## 4.5. Подготовка к промежуточной аттестации в форме экзамена

В качестве методических материалов при подготовке к экзамену рекомендуется использовать собственные конспекты лекций и источники, рекомендованные в списке литературы раздела 7.

Контрольные вопросы и задания для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведены в п. 5.2.

## 5. Фонд оценочных средств для промежуточной аттестации по дисциплине (модулю), включающий:

### 5.1. Описание шкал оценивания результатов обучения по дисциплине

Уровень сформированности компетенций (индикатора достижения компетенций)	Шкала оценивания сформированности 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, без ошибок.	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.	Продemonстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в полном объеме.	Продemonстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме, но некоторые с недочетами.	Продemonстрированы все основные умения, решены все основные задачи с небольшими недочетами, выполнены все задания в полном объеме.	Продemonстрированы все основные умения, решены все основные задачи. Выполнены все задания, в полном объеме без недочетов
<u>Навыки</u>	Отсутствие владения материалом. 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.	Продemonстрированы базовые навыки при решении стандартных задач с некоторыми недочетами	Продemonстрированы базовые навыки при решении стандартных задач без ошибок и недочетов.	Продemonстрированы навыки при решении нестандартных задач без ошибок и недочетов.	Продemonстрирован творческий подход к решению нестандартных задач.

### Шкала оценки при промежуточной аттестации

Оценка	Уровень подготовки
--------	--------------------

зачтено	Превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно»
	Отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
	Очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
	Хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «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», ни одна из компетенций не сформирована на уровне «плохо»
	Плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

## 5.2. Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения

### 5.2.1. Типовые задания для оценки сформированности компетенции ПК-4

- Дано следующее определение.  

```
(define p (lambda () (p)))
```

Что произойдет при вычислении  $p$ ,  $(p)$ ,  $((p))$ ?  
 Если вычисление возвращает результат, напишите его, а также его тип.
- Рассмотрим следующее определение.  

```
(define lst '(() (((1 a) 2) 3 4) 5))
```

Напишите выражение, содержащее только `car`, `cdr` и `lst` и возвращающее символ `a`.
- 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))`.
- Рассмотрим следующее определение.  

```
(define (f a)
  (lambda (g)
    (+ a ((g (* 2 a)) (lambda (x) (lambda (h) (* 3 x)))))))
```

Напишите лямбда-терм, соответствующий функции `f`. Напишите цепочку  $\beta$ -редукций, начинающуюся с `(f 1)` `f` и заканчивающуюся числом.

### 5.2.2. Темы программных проектов для оценки сформированности компетенции ПК-11

- Выражение  

```
(while test
  exp1
  ...
  expn)
```

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 `exp1`, ..., `expn`, после чего снова проверяется `test` и т.д. Напишите функцию `transform-while`, переводящую `while` в следующее выражение.

```
(let ()
  (define (loop)
    (if (not test)
        (begin
          exp1
          ...
          expn
          (loop))
        (void)))
  (loop))
```

Добавьте форму `while` к интерпретатору в качестве синтаксического расширения.

2. Выражение

```
(until
  exp1
  ...
  expn
  test)
```

последовательно вычисляются `exp1`, ..., `expn`, после чего вычисляет значение `test`. Если оно ложно (то есть равно `#f`), то снова вычисляются значения `exp1`, ..., `expn` и т.д. Если значение `test` истинно (отлично от `#f`), вычисление заканчивается с неопределенным значением `#void`. Проверьте работу `until` на примерах.

Напишите функцию `transform-until`, переводящую особую форму `until` в следующее выражение.

```
(let ()
  (define (loop)
    exp1
    ...
    expn
    (if test (void) (loop)))
  (loop))
```

Добавьте форму `until` к интерпретатору в качестве синтаксического расширения.

3. Добавьте в ядро интерпретатора особую форму `for/or`. Она работает аналогично `for`, но в отличие от `for` цикл работает до тех пор, пока значением последнего выражения в теле цикла является `#f`. Если значение `v` последнего выражения отлично от `#f`, цикл заканчивается и значением всего выражения становится `v`. Если тело цикла не вычисляется ни одного раза или если его значение равно `#f` до окончания итераций, значением всего выражения становится `#f`.

Примеры:

```
> (for/or ([i '(1 2 3 4 x)]) (> i 3))
#t
> (for/or ([i '(1 2 3 4 x)]) i)
1
> (for/or ([i '()]) (/ 1 0))
#f
```

### 5.2.3. Контрольные вопросы

Вопрос	Код компетенции (согласно РПД)
1. Использование нехвостостовой рекурсии и хвостовой рекурсии (итерации) в языках функционального программирования. Их преимущества и недостатки. Примеры.	ПК-4
2. Функции высших порядков. Примеры: map, filter, левая и правая свертки.	ПК-4
3. Основная теорема о рекуррентных соотношениях. Определение временной и пространственной сложности функций.	ПК-4
4. Присваивание в Racket. Списочные диаграммы.	ПК-4
5. Лямбда-исчисление: определение термов и $\beta$ -редукции. Примеры вычисления лямбда-термов.	ПК-4
6. Модель вычисления с окружениями. Необходимость замыкания функций.	ПК-11
7. Множественные представления абстрактных данных. Барьеры абстракции. Множественные представления для абстрактных данных.	ПК-4
8. Параллелизм, механизмы управления параллелизмом	ПК-11
9. Потоки и задержанное вычисление	ПК-11
10. Нормальный и аппликативный порядок вычислений. Интерпретатор с ленивым вычислением	ПК-11
11. Недетерминированные вычисления. Реализация интерпретатора с недетерминированными вычислениями с использованием продолжений	ПК-11
12. Логическое программирование. Унификация	ПК-11
13. Регистровые машины. Реализация рекурсии с помощью стека	ПК-11
14. Выделение памяти и сборка мусора	ПК-11
15. Компиляция языка функционального программирования	ПК-11

## 6. Учебно-методическое и информационное обеспечение дисциплины

### а) основная литература:

1. Кубенский, А. А. Функциональное программирование : учебник и практикум для вузов / А. А. Кубенский. — Москва : Издательство Юрайт, 2022. — 348 с. — (Высшее образование). — ISBN 978-5-9916-9242-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/490015> (дата обращения: 19.09.2022).
2. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2022. — 320 с. — (Высшее образование). — ISBN 978-5-534-02444-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489754> (дата обращения: 19.09.2022).

### б) дополнительная литература:

Курс «Введение в теорию программирования. Функциональный подход»: <https://intuit.ru/studies/courses/39/39/info>.

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

Racket Programming Language: <https://racket-lang.org/>.



## **7. Материально-техническое обеспечение дисциплины**

Помещения представляют собой учебные аудитории для проведения учебных занятий, предусмотренных программой (лекционного и семинарского типа), оснащенные оборудованием и техническими средствами обучения.

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

Программа составлена в соответствии с требованиями ОС ВО ННГУ с учетом рекомендаций ФГОС ВО по направлению 01.04.02 Прикладная математика и информатика.

Автор: Ph.D., доц. \_\_\_\_\_ Е.М. Макаров

Рецензент: \_\_\_\_\_

Зав. кафедрой: д.ф.м.н., доц. \_\_\_\_\_ Н.Ю. Золотых

Программа одобрена на заседании методической комиссии института информационных технологий, математики и механики от «30» ноября 2022 г. № 13