



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.
Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храмшин

03.02.2026 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ

Направление подготовки (специальность)
09.03.01 Информатика и вычислительная техника

Направленность (профиль/специализация) программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования - бакалавриат

Форма обучения
очная

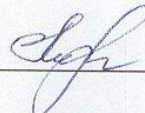
| | |
|---------------------|---|
| Институт/ факультет | Институт энергетики и автоматизированных систем |
| Кафедра | Вычислительной техники и программирования |
| Курс | 3 |
| Семестр | 5 |

Магнитогорск
2026 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 929)

Рабочая программа рассмотрена и одобрена на заседании кафедры
Вычислительной техники и программирования
29.01.2026, протокол № 7

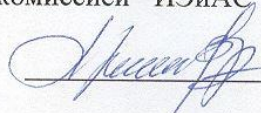
Зав. кафедрой



О.С. Логунова

Рабочая программа одобрена методической комиссией ИЭиАС
03.02.2026 г. протокол № 5

Председатель



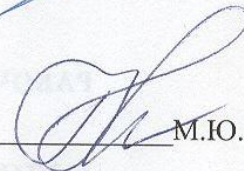
В.Р. Храмшин

Рабочая программа составлена:
Старший преподаватель кафедры ВТиП,



Зарецкий М.В.

Рецензент:
Директор НИИ "Промбезопасность", д-р техн. наук



М.Ю. Наркевич

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2027 - 2028 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2028 - 2029 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2029 - 2030 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2030 - 2031 учебном году на заседании кафедры Вычислительной техники и программирования

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ О.С. Логунова

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) «Функциональное программирование» являются:

- формирование у студентов понимания роли функциональной парадигмы программирования в теории и практике разработки, сопровождения и эксплуатации программного обеспечения;
- выработка умения применять технологии функционального программирования для решения практических задач.
- освоение взаимосвязей функциональной и объектно-ориентированной парадигм программирования;
- освоение современных методов проектирования программных продуктов на основе функциональной парадигмы.

Для достижения поставленных целей в курсе «Функциональное программирование» решаются задачи:

- изучение языка функционального программирования (LISP с учетом диалектов);
- изучение функциональных расширений современных языков программирования (Python, JavaScript, встроенный язык Matlab);
- изучение современных применений функциональной парадигмы программирования.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Функциональное программирование входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Информатика

Численные методы

Программирование

Прикладная математика

Философия

Математическая логика и дискретная математика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Подготовка к сдаче и сдача государственного экзамена

Технологии Data Mining и Big Data

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Функциональное программирование» обучающийся должен обладать следующими компетенциями:

| Код индикатора | Индикатор достижения компетенции |
|----------------|---|
| ПК-2 | Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам |
| ПК-2.1 | Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для автоматизированных систем |

| | |
|--|---|
| ПК-6 Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями | |
| ПК-6.1 | Оценивает качество математической модели при формализации задачи предметной области |
| ПК-6.2 | Оценивает качество разработанных алгоритмов для последующего кодирования |
| ПК-6.3 | Оценивает выбор программных средств для программирования и манипулирования данными в соответствии установленными требованиями |

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 академических часов, в том числе:

- контактная работа – 57,2 академических часов;
- аудиторная – 54 академических часов;
- внеаудиторная – 3,2 академических часов;
- самостоятельная работа – 51,1 академических часов;
- в форме практической подготовки – 0 академических часов;
- подготовка к экзамену – 35,7 академических часов

Форма аттестации - экзамен

| Раздел/ тема дисциплины | Семестр | Аудиторная контактная работа (в академических часах) | | | Самостоятельная работа студента | Вид самостоятельной работы | Форма текущего контроля успеваемости и промежуточной аттестации | Код компетенции |
|---|---------|--|-----------|-------------|---------------------------------|--|---|-----------------|
| | | Лек. | лаб. зан. | практ. зан. | | | | |
| 1. Функциональная парадигма программирования | | | | | | | | |
| 1.1 Программа как суперпозиция функций. Рекурсивные функции и λ -исчисление А. Черча. Программирование в функциональных обозначениях. Редукция и ее виды. Редексы. Нормальный порядок редукции. Аппликативный порядок редукции. Теорема Черча Россера | 5 | 2 | 2 | | 2 | Самостоятельное изучение учебной и научной литературы | Беседа – обсуждение. Устный опрос. | ПК-2.1 |
| 1.2 Функциональные языки. Строго функциональный язык. Функциональные компоненты нефункциональных языков программирования | | 2 | 2 | | 2 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы. | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-2.1, ПК-6.2 |
| Итого по разделу | | 4 | 4 | | 4 | | | |
| 2. Работа со списками | | | | | | | | |
| 2.1 Основные понятия языка LISP. Атомы, точечные пары, списки. S-выражения. Функции в LISP. Данные в языке LISP. Типы данных. Арифметические | 5 | 2 | 4 | | 4 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-6.1, ПК-6.3 |

| | | | | | | | | |
|---|---|---|----|--|------|--|--|----------------|
| функции, логические функции, функции работы со строковыми данными. | | | | | | занятию. Выполнение лабораторной работы. | | |
| 2.2 Функции для работы со списками и их суперпозиции в языке LISP. Функции работы со списками в языке Python. Глобальные и локальные переменные в языках LISP и Python. | 5 | 2 | 4 | | 4 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы. | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-6.2, ПК-6.3 |
| Итого по разделу | | 4 | 8 | | 8 | | | |
| 3. Рекурсия. Ассоциативные списки. | | | | | | | | |
| 3.1 Виды рекурсии. Реализация рекурсии в языках LISP и Python. | 5 | 2 | 6 | | 6 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы. | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-2.1, ПК-6.1 |
| 3.2 Ассоциативные списки в языке LISP. Словари в языке Python. Функции для ассоциативного поиска. Hash – таблицы в языках LISP и Python. Анонимные функции в языках LISP, Python, встроенном языке Matlab / Octave. | | 2 | 6 | | 8 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы. | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-6.1, ПК-6.2 |
| Итого по разделу | | 4 | 12 | | 14 | | | |
| 4. Функционалы. Классы и объекты. Практические реализации. | | | | | | | | |
| 4.1 Функционалы в языках LISP и Python. Виды функционалов. Совместное применение функционалов и анонимных функций. | 5 | 2 | 6 | | 8 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному занятию. Выполнение лабораторной работы. | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-6.1, ПК-6.3 |
| 4.2 классы и объекты в языках LISP и Python. Синтез функциональной и объектно-ориентированной парадигм программирования. | | 4 | 6 | | 17,1 | Самостоятельное изучение учебной и научной литературы. Подготовка к лабораторному | Беседа – обсуждение. Анализ программного кода. Устный опрос. | ПК-6.2, ПК-6.3 |

| | | | | | | | | |
|--|---|----|----|--|------|---|-----------------------------|--------------------------------|
| Диалект Visual LISP для систем компьютерной графики AutoCAD и NanoCAD. | | | | | | занятию. Выполнение лабораторной работы. | | |
| Итого по разделу | | 6 | 12 | | 25,1 | | | |
| 5. Контроль | | | | | | | | |
| 5.1 Внеаудиторная контактная работа. | 5 | | | | | Анализ программного кода. Устранение ошибок. | Проверка программного кода. | ПК-2.1, ПК-6.1, ПК-6.2, ПК-6.3 |
| 5.2 Подготовка к экзамену. | | | | | | Решение комплексных задач. | Проверка программного кода | ПК-2.1, ПК-6.1, ПК-6.2, ПК-6.3 |
| Итого по разделу | | | | | | | | |
| Итого за семестр | | 18 | 36 | | 51,1 | | экзамен | |
| Итого по дисциплине | | 18 | 36 | | 51,1 | | экзамен | |

5 Образовательные технологии

1. Традиционные образовательные технологии ориентируются на организацию образовательного процесса, предполагающую прямую трансляцию знаний от преподавателя к студенту (преимущественно на основе объяснительно-иллюстративных методов обучения). Учебная деятельность студента носит в таких условиях, как правило, репродуктивный характер.

Формы учебных занятий с использованием традиционных технологий:

Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).

Семинар – беседа преподавателя и студентов, обсуждение заранее подготовленных сообщений по каждому вопросу плана занятия с единым для всех перечнем рекомендуемой обязательной и дополнительной литературы.

Практическое занятие, посвященное освоению конкретных умений и навыков по предложенному алгоритму.

Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.

2. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.

3. Интерактивные технологии – организация образовательного процесса, которая предполагает активное и нелинейное взаимодействие всех участников, достижение на этой основе лично значимого для них образовательного результата. Наряду со специализированными технологиями такого рода принцип интерактивности прослеживается в большинстве современных образовательных технологий. Интерактивность подразумевает субъект - субъектные отношения в ходе образовательного процесса и, как следствие, формирование саморазвивающейся информационно-ресурсной среды.

Формы учебных занятий с использованием специализированных интерактивных технологий:

Лекция «обратной связи» – лекция–провокация (изложение материала с заранее запланированными ошибками), лекция-беседа, лекция-дискуссия, лекция–пресс-конференция.

4. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 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, 2026. — 348 с. —

(Высшее образование). — ISBN 978-5-9916-9242-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/583854> (дата обращения: 08.03.2026).

б) Дополнительная литература:

1. Петренко, А. А. Функциональное программирование : учебное пособие / А. А. Петренко, А. О. Суворов, Н. Ф. Шаякбаров. — Пермь : ПНИПУ, 2022. — 160 с. — ISBN 978-5-398-02797-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/328847> (дата обращения: 10.05.2023). — Режим доступа: для авториз. пользователей.

в) Методические указания:

1. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 164 с. — (Высшее образование). — ISBN 978-5-534-00844-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/512894> (дата обращения: 03.05.2023).

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

| Наименование ПО | № договора | Срок действия лицензии |
|--|---|------------------------|
| Deductor Studio Academic | Соглашение о сотрудничестве №06-2901\08 от 29.01.2008 | бессрочно |
| Anaconda Python | свободно распространяемое ПО | бессрочно |
| MathWorks MathLab v.2014 Classroom License | К-89-14 от 08.12.2014 | бессрочно |
| NotePad++ | свободно распространяемое ПО | бессрочно |
| Autodesk AutoCad 2011 Master Suite | К-526-11 от 22.11.2011 | бессрочно |

Профессиональные базы данных и информационные справочные системы

| Название курса | Ссылка |
|---|---|
| Электронная база периодических изданий ООО «ИВИС» | https://eivis.ru/ |

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Лекционная аудитория ауд. 282 – Мультимедийные средства хранения, передачи и представления информации;

Компьютерные классы Центра информационных технологий ФГБОУ ВПО «МГТУ им. Г.И. Носова» – Персональные компьютеры, объединенные в локальные сети с выходом в Internet, оснащенные современными программно-методическими комплексами для решения задач в области информатики и вычислительной техники;

Аудитории для самостоятельной работы: компьютерные классы; читальные залы библиотеки – ауд. 282 и классы УИТ и АСУ;

Помещения для самостоятельной работы обучающихся, оснащенных компьютерной техникой с возможностью подключения к сети «Интернет» и наличием доступа в электронную информационно-образовательную среду организации – классы УИТ и АСУ;

Помещения для хранения и профилактического обслуживания учебного оборудования – Центр информационных технологий – ауд. 372

Приложение 1.

Учебно-методическое обеспечение самостоятельной работы обучающихся

В течение семестра каждый студент выполняет лабораторные работы.

Лабораторная работа №1. Программа как суперпозиция функций. Рекурсивные функции и λ -исчисление А. Черча. Программирование в функциональных обозначениях. Редукция и ее виды. Редексы. Нормальный порядок редукции. Аппликативный порядок редукции.

Теорема Черча – Россера

Задание 1 (пороговый уровень).

Задана функция $f(x) = e^{(x+3)(x-9)^2}$ Ее аргумент является функцией $x = x(t) = \sqrt{t} \cdot \ln^2(3t)$

1.1. Запишите суперпозицию функций в префиксной нотации.

1.2. Запишите вызовы встроженных функций в виде аппликации.

Задание 2 (пороговый уровень). Для заданной в предыдущем пункте суперпозиции функций определите редексы.

2.1 Выполните редукцию в аппликативном порядке.

2.2 Выполните редукцию в нормальном порядке.

Задание 3(пороговый уровень) Задайте допустимый числовой аргумент

3.1. Определите количество шагов, необходимых для завершения редукции в аппликативном порядке.

3.2. Определите количество шагов, необходимых для завершения редукции в нормальном порядке.

Задание 4 (средний уровень)

4.1. Выпишите представленные функции в нотации λ -исчисления.

4.2. Проиллюстрируйте на примере представленных функций теорему Черча-Россера.

Задание 5 (высокий уровень)

5.1. Реализуйте программную реализацию представленных функций средствами языка Common Lisp.

5.2. Реализуйте программную реализацию представленных функций средствами языка Python.

5.3. Сгенерируйте аналогичный программный код с помощью нейросетей GigaChat и Deepseek

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №2. Функциональные языки. Строго функциональный язык.

Функциональные компоненты нефункциональных языков программирования

Дана программа на языке Python.

```
def myProg(arr):
```

```
k=len(arr)
for i in range(k):
    arr[i]*=2
print (arr)
```

Задание 1 (пороговый уровень)

- 1.1. Запишите возможные типы для передаваемого аргумента.
- 1.2. Запишите обращение к функции определения длины в виде аппликации.

Задание 2 (пороговый уровень)

- 2.1. Замените используемый в программе цикл на рекурсию:
- 2.2. Определите критерий останова рекурсии.

Задание 3 (средний уровень)

- 3.1. Перепишите программу в виде суперпозиции редексов.
- 3.2. Выполните для заданной суперпозиции редукцию в нормальном порядке.
- 3.3. Выполните для заданной суперпозиции редукцию в аппликативном порядке

Задание 4 (высокий уровень)

- 4.1. Выполните программную реализацию для заданной суперпозиции средствами языка Common LISP.
- 4.2. Сгенерируйте аналогичный программный код с помощью нейросетей GigaChat и Deepseek.
- 4.3 Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №3. Основные понятия языка LISP. Атомы, точечные пары, списки. S-выражения. Функции в LISP. Данные в языке LISP. Типы данных. Арифметические функции, логические функции, функции работы со строковыми данными. Для выполнения всех заданий используем язык программирования GNU Common LISP.

Задание 1(пороговый уровень)

- 1.1. Создайте по одному атому всех известных вам типов данных
- 1.2. Из всех созданных атомов создайте точечные пары

Задание 2 (пороговый уровень)

- 2.1. Создайте из одного атома с помощью функции **cons** одноэлементный список.
- 2.2. Создайте из нескольких атомов с помощью рекуррентного применения функции **cons** многоэлементный список.

Задание 3 (пороговый уровень)

- 3.1. Создайте список, предназначенный для выполнения арифметической операции над совокупностью аргументов.
- 3.2. Создайте список, предназначенный для выполнения логической (булевой) операции над совокупностью аргументов.

Задание 4 (средний уровень)

- 4.1. Составьте функцию с двумя параметрами (атомами) для выполнения арифметической операции.
- 4.2. Составьте функцию с двумя обязательными и одним необязательным параметром для выполнения арифметической операции.

Задание 5 (высокий уровень)

- 5.1. Составьте функцию для работы со списком, содержащим разнотипные данные.
- 5.2. Сгенерируйте аналогичную функцию с помощью нейросетей GigaChat и Deepseek.
- 5.3. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №4. Функции работы со списками и их суперпозиции в языке LISP. Функции работы со списками в языке Python. Глобальные и локальные переменные в языках LISP и Python.

Задание 1(пороговый уровень)

- 1.1. С помощью функции **car** извлеките голову заданного списка

1.2. С помощью функции **cdr** извлеките голову заданного списка

1.3. Выполните аналогичные действия средствами Python.

Задание 2(пороговый уровень)

2.1. С помощью суперпозиций функций **car** и **cdr** выделите заданный элемент списка.

2.2. Выполните аналогичные действия средствами Python.

Задание 3 (средний уровень)

3.1. Средствами языка LISP создайте глобальные переменные.

3.2. Выполните аналогичные действия средствами Python.

Задание 4 (средний уровень)

4.1. Средствами языка LISP создайте локальную переменную.

4.2. Выполните аналогичные действия средствами Python.

Задание 5 (высокий уровень)

5.1. Разработайте программу на языке LISP, в которой имеется несколько функций, используются глобальные и локальные переменные.

5.2. Разработайте аналогичную программу на языке Python.

5.3. Сгенерируйте аналогичные программы с помощью нейросетей GigaChat и Deepseek.

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №5. Виды рекурсии. Реализация рекурсии в языках LISP и Python.

Задание 1(пороговый уровень)

1.1 Средствами языка LISP запрограммируйте вычисление факториала с помощью рекурсивной функции.

1.2 То же задание выполните средствами языка Python.

Задание 2(пороговый уровень)

2.1. Средствами языка LISP запрограммируйте вычисление n -го числа Фибоначчи.

2.2. То же задание выполните средствами языка Python.

Задание 3(пороговый уровень)

3.1. Средствами языка LISP запрограммируйте преобразование списка с подписками в одноуровневый список.

3.2. То же задание выполните средствами языка Python.

Задание 4 (средний уровень)

4.1. Средствами языка LISP запрограммируйте поиск максимального числового элемента в списке с подписками.

4.2. То же задание выполните средствами языка Python.

Задание 5 (высокий уровень)

5.1. Разработайте программу на языке LISP, которая находит на заданном интервале корень нелинейного уравнения методом простой итерации.

5.2. Разработайте аналогичную программу на языке Python.

5.3. Сгенерируйте аналогичные программы с помощью нейросетей GigaChat и Deepseek.

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №6. Ассоциативные списки в языке LISP. Словари в языке Python.

Функции для ассоциативного поиска. Hash – таблицы в языке LISP.

Задание 1(пороговый уровень)

1.1. Средствами языка LISP создайте ассоциативный список, содержащий не менее семи пар «ключ-значение».

1.2. То же задание выполните средствами языка Python (в данном случае должен быть создан словарь).

Задание 2(пороговый уровень)

2.1. Средствами языка LISP запрограммируйте поиск по ассоциативному списку.

2.2. Средствами языка Python запрограммируйте поиск по словарю.

Задание 3(пороговый уровень)

3.1. Средствами языка LISP запрограммируйте поиск по Hash-таблице.

3.2. Сопоставьте эффективность поиска по Hash-таблице и словарю.

Задание 4(средний уровень)

4.1. Спроектируйте и реализуйте средствами языка LISP Hash-таблицу для хранения и обработки разнотипных данных.

4.2. Спроектируйте и реализуйте средствами языка Python словарь для той же цели.

Задание 5 (высокий уровень)

5.1. Разработайте программу на языке LISP, которая выполняет создание, обновление и выборку данных по определенному критерию для Hash-таблицы.

5.2. Разработайте аналогичную программу для словаря на языке Python.

5.3. Сгенерируйте аналогичные программы с помощью нейросетей GigaChat и Deepseek.

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №7. Функционалы в языках LISP и Python. Виды отображающих функционалов. Совместное применение функционалов и анонимных функций.

Задание 1(пороговый уровень)

1.1. С помощью отображающего функционала запрограммируйте поэлементное перемножение двух числовых одноуровневых списков одинаковой длины.

1.2. То же задание выполните средствами языка Python.

1.3. Сопоставьте результат с результатом поэлементного умножения двух numpy-массивов в Python.

Задание 2(пороговый уровень)

2.1. Выполните выборку из списка, содержащего разнотипные элементы, с помощью отображающего функционала и анонимной функции.

2.2. То же задание выполните средствами языка Python.

Задание 3(пороговый уровень)

3.1. С помощью отображающего функционала и анонимной функции запрограммируйте средствами языка LISP построение линейной комбинации нескольких списков, содержащих числовые элементы.

3.2. То же задание выполните средствами языка Python.

Задание 4(средний уровень)

4.1. С помощью отображающего функционала и анонимной функции запрограммируйте средствами языка LISP выборку элементов из Hash-таблицы по сложному критерию.

4.2. То же задание выполните средствами языка Python (для словаря).

Задание 5 (высокий уровень)

5.1. Разработайте программу на языке LISP, которая, используя отображающий функционал и анонимную функцию выполняет проверку соответствия нескольких сложных списков набору логических критериев.

5.2. Разработайте аналогичную программу для на языке Python.

5.3. Сгенерируйте аналогичные программы с помощью нейросетей GigaChat и Deepseek.

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Лабораторная работа №8. Синтез функциональной и объектно-ориентированной парадигм программирования. Диалект AutoLISP для системы компьютерной графики NanoCAD.

Задание 1(пороговый уровень)

1.1. Создайте средствами языка программирования LISP структуру, содержащую несколько разнотипных полей.

1.2. Разработайте программу для генерации, редактирования структур средствами языка LISP.

Задание 2(пороговый уровень)

2.1. Средствами языка LISP создайте класс, содержащий конструктор, деструктор, методы добавления, удаления, редактирования записи.

2.2. То же задание выполните средствами языка Python.

2.3. Сформулируйте принципиальные различия в реализации объектной парадигмы в языках LISP и Python.

Задание 3(пороговый уровень)

3.1. Средствами языка VisualLISP напишите программу для автоматизации построения несложного чертежа в среде NanoCAD.

3.2. С помощью объектных свойств VisualLISP получите сведения о площади фигуры, построенной в среде NanoCAD.

Задание 4(средний уровень)

4.1. Средствами языка LISP в ранее созданном классе создайте метод, реализующий поиск данных по сложному набору критериев.

4.2. То же задание выполните средствами языка Python.

Задание 5 (высокий уровень)

5.1 Разработайте программу на языке LISP, которая реализует множественное наследование.

5.2. Разработайте аналогичную программу для на языке Python.

5.3. Сгенерируйте аналогичные программы с помощью нейросетей GigaChat и Deepseek.

5.4. Сравните полученный каждым из способов программный код по критериям: правильность, удобочитаемость, приспособленность к доработке.

Приложение 2. Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|---|---|--|
| ПК-2: Способность к анализу проблемной ситуации, разработке требований к системе, постановке целей создания системы, разработке концепции системы и технического задания на создание системы, представления концепции, технического задания на систему и изменений в них заинтересованным лицам | | |
| ПК-2.1: | Оценивает выбор средств и методов для проведения системного анализа при проектировании программного обеспечения для автоматизированных систем | <i>Перечень теоретических вопросов</i> <ol style="list-style-type: none">1. Понятие об императивном и декларативном программировании.2. Функция, как отображение.3. Программа, как суперпозиция функций.4. λ - выражения. Свободные и связанные переменные5. Понятие о редукции и редексе.6. Аппликативный порядок редукции.7. Нормальный порядок редукции.8. Теорема Черча-Россера.9. Энергичные и ленивые вычисления. <i>Задания на решение задач из профессиональной области, комплексные задания</i> Задание 1. <ol style="list-style-type: none">1.1. Рассмотрите натуральный логарифм от многочлена, как функцию, отображающую вещественные числа на множество вещественных чисел. Какие потребуются ограничения для аргумента?1.2. Рассмотрите натуральный логарифм от многочлена, как функцию, отображающую вещественные числа на множество комплексных |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|--|--|--|
| | | <p>чисел. Как изменятся условия выполнимости отображения по сравнению с предыдущим вариантом?</p> <p>Задание 2.</p> <p>2.1. Рассмотрите множество аффинных преобразований на плоскости. Определите, из какого множества в какое множество выполняется преобразование.</p> <p>2.2. Рассмотрите полином, задающий кривую Безье, как функцию, определяющую преобразование точек плоскости. Сформулируйте алгоритм построения кривой Безье в виде суперпозиции нелинейных преобразований.</p> |
| <p>ПК-6: Способность к формализации и алгоритмизации поставленных задач, к написанию программного кода с использованием языков программирования, определения и манипулирования данными и оформлению программного кода в соответствии установленными требованиями</p> | | |
| ПК-6.1 | <p>Оценивает качество математической модели при формализации задачи предметной области</p> | <p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Целочисленные данные в Common LISP и определенные для них операции. 2. Вещественные данные в Common LISP и определенные для них операции. 3. Комплексные данные в Common LISP и определенные для них операции. 4. Логические данные Common LISP и определенные для них операции. 5. Точечные пары. Построение точечной пары. Операции с точечными парами. 6. Списки и атомы. Основные понятия. 7. Функции QUOTE и EVAL 8. Основные функции для работы со списками. (CAR, CDR и их суперпозиции, nth 9. Функции SETQ и SETF в Common LISP. 10. Определение пользовательской функции в Common LISP. Виды передаваемых в функцию параметров. 11. Возвращение значений из пользовательской функции в Common LISP. <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Задание 1.</p> <p>1.3. Постройте многоэлементный список, рекурсивно применяя функцию построения точечных пар.</p> <p>1.4. Выделите голову списка.</p> <p>1.5. Выделите голову хвоста списка.</p> <p>Задание 2.</p> <p>2.1. Разработайте функцию для вычисления количества перестановок, сочетаний и размещений без повторений.</p> <p>Разработайте аналогичную функцию с помощью</p> |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|--|--|
| | | <p>ИИ-систем. Сопоставьте результаты.</p> <p>2.2. Разработайте функцию для нахождения количества перестановок, сочетаний и размещений с повторениями. Разработайте аналогичную функцию с помощью ИИ-систем. Сопоставьте результаты.</p> |
| ПК-6.2 | Оценивает качество разработанных алгоритмов для последующего кодирования | <p><i>Перечень теоретических вопросов</i></p> <ol style="list-style-type: none"> 1. Массивы в Common LISP. Создание массива. 2. Функции для работы с массивами в Common LISP. 3. Особенности работы с динамическими массивами в Common LISP. 4. Создание глобальных переменных в Common LISP. 5. Создание констант в Common LISP. 6. Создание локальных переменных в Common LISP. 7. LET- и LET*- области, их вложенность. 8. Управление ходом вычислений в Common LISP. Функция COND.. 9. Управление ходом вычислений в Common LISP. Функции IF, WHEN, UNLESS. 10. Циклы в Common LISP. 11. Ассоциативные списки и их использование. 12. Функционалы в Common LISP. 13. Использование анонимных функций при работе с функционалами. 14. Рекурсия. Виды рекурсии 15. Структуры в Common LISP. 16. Функции для эффективной работы со структурами в Common LISP. <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Задание 1.</p> <ol style="list-style-type: none"> 1.1. Создайте структуру, содержащую разнотипные поля. 1.2. Список структур созданного типа. 1.3. Создайте функции манипуляции элементами списка структур 1.4. Реализуйте функционирование системы. 1.5. Разработайте аналогичную функцию с помощью ИИ-систем. Сопоставьте результаты. <p>Задание 2.</p> <ol style="list-style-type: none"> 2.1. Средствами Common LISP постройте информационную систему, основанную на использовании Hash-таблиц.. 2.2. Реализуйте в данной системе функции ввода, корректировки, удаления, поиска информации.. 2.3. Разработайте аналогичную функцию с помощью ИИ-систем. Сопоставьте результаты. |
| ПК-6.3 | Оценивает выбор программных средств для программирования и | <ol style="list-style-type: none"> 1. Анонимные функции в Common LISP. 2. Использование анонимных функций при работе с функционалами. |

| Код индикатора | Индикатор достижения компетенции | Оценочные средства |
|----------------|--|--|
| | манипулирования данными в соответствии установленными требованиями | <p>3. Классы и объекты в Common LISP. 4. Создание методов. Особенности методов в Common LISP. 5. Постметоды в Common LISP. 6. Инкапсуляция в Common LISP. 7. Единичное и множественное наследование в Common LISP. 8. Полиморфизм в Common LISP. 9. Особенности AutoLISP. Среда разработки в NanoCAD. 10. Специальные геометрические функции в AutoLISP 11. Специфика создания и использования глобальных переменных в AutoLISP 12. Специфика создания и использования локальных переменных в AutoLISP 13. Специфика функций ввода-вывода в AutoLISP 14. Объектная модель AutoLISP. 15. Объектные свойства геометрических примитивов Nanocad и их получение.</p> <p><i>Задания на решение задач из профессиональной области, комплексные задания</i></p> <p>Задание 1. 1.1. Создайте класс, содержащий явно прописанные конструктор, деструктор, методы для изменения значений полей. 1.2. Создайте не менее двух классов-наследников для данного класса. 1.3. Пользуясь множественным наследованием, создайте класс-наследник двух наследников 1.4. Реализуйте функционирование системы. 1.5. Разработайте аналогичную функцию с помощью ИИ-систем. Сопоставьте результаты.</p> <p>Задание 2. 2.1. Средствами AutoLISP постройте в среде NanoCAD замкнутую линию сложной формы. 2.2. Пользуясь объектными средствами AutoLISP определите площадь построенной фигуры. 2.3. Разработайте аналогичную функцию с помощью ИИ-систем. Сопоставьте результаты.</p> |

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Функциональное программирование» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме экзамена.

Экзамен по дисциплине проводится по результатам отчетности за выполненные самостоятельные работы с опросом в устной форме по этапам выполнения в беседе-обсуждении на лекционных занятиях.

Критерии оценки

- на оценку **«отлично»** – полно раскрыто содержание материала; чётко и правильно даны определения и раскрыто содержание материала; ответ самостоятельный, при ответе использованы знания, приобретённые ранее;
- на оценку **«хорошо»** – раскрыто основное содержание материала в объёме; в основном правильно даны определения, понятия; материал изложен неполно, при ответе допущены неточности, нарушена последовательность изложения; допущены небольшие неточности при выводах и использовании терминов; практические навыки нетвёрдые;
- на оценку **«удовлетворительно»** – усвоено основное содержание материала, но изложено фрагментарно, не всегда последовательно; определения и понятия даны не чётко; практические навыки слабые;
- на оценку **«неудовлетворительно»** – основное содержание учебного материала не раскрыто; не даны ответы на дополнительные вопросы преподавателя