

Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(государственный университет)»



УТВЕРЖДАЮ

Проректор по учебной работе
и довузовской подготовке

А.А. Воронов

Рабочая программа дисциплины (модуля):
по дисциплине: Сравнительный анализ языков программирования
по направлению: Компьютерная безопасность (специалитет)
профиль подготовки: Безопасность распределенных компьютерных систем
факультет радиотехники и кибернетики
кафедра информатики и вычислительной математики
курс: 2
квалификация: специалист

Семестры, формы промежуточной аттестации:

3(Осенний) - без промежуточной аттестации

4(Весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 60 час.

практические и семинарские занятия: 0 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 120 час.

Всего часов: 180, всего зач. ед.: 5

Количество курсовых работ, заданий: 8

Программу составил: А.В. Пензин, старший преподаватель

Программа обсуждена на заседании кафедры

15 марта 2018 г.

СОГЛАСОВАНО:

Заведующий кафедрой

И.Б. Петров

Начальник учебного управления

И.Р. Гарайшина

Декан факультета

С.Н. Гаричев

1. Цели и задачи

Цель дисциплины

изучение современных подходов в реализации языков программирования, сравнительный анализ языков программирования, решение алгоритмических задач распознавания, порождения и построения в современных языках программирования.

Задачи дисциплины

- освоение студентами базовых знаний в области сравнительного анализа языков программирования (САЯП);
- приобретение теоретических знаний в области анализа языков программирования и основных подходов к анализу алгоритмов построенных на различных языках;
- оказание консультаций и помощи студентам в проведении собственных теоретических и экспериментальных исследований в области САЯП;
- формирование навыков применения САЯП при исследовании экспериментальных или экспертных данных при выполнении студентами выпускных работ на степень бакалавра.

2. Место дисциплины (модуля) в структуре образовательной программы

Данная дисциплина относится к базовой части ООП.

Дисциплина «Сравнительный анализ языков программирования» базируется на дисциплинах:

Дифференциальные уравнения;
Математический анализ;
Высшая математика.

Дисциплина «Сравнительный анализ языков программирования» предшествует изучению дисциплин:

Прикладные физико-технические и компьютерные методы исследований.

3. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Освоение дисциплины направлено на формирование следующих общекультурных, общепрофессиональных и профессиональных компетенций:

- способность к самоорганизации и самообразованию (ОК-8);
- способность к самостоятельному построению алгоритма, проведению его анализа и реализации в современных программных комплексах (ОКП-10);
- способность применять методы научных исследований в профессиональной деятельности, в том числе в работе над междисциплинарными и инновационными проектами (ОПК-4);
- способность учитывать современные тенденции развития информатики и вычислительной техники, компьютерных технологий в своей профессиональной деятельности, работать с программными средствами общего и специального назначения (ОПК-7).

В результате освоения дисциплины обучающиеся должны

знать:

- фундаментальные понятия и методы теории алгоритмов;
- современные проблемы анализа алгоритмов, постановки задачи, выбора языка программирования, построения полного решения задачи;
- методы и подходы решения практических алгоритмических задач, от постановки задачи до финального алгоритма;
- инструментальные языковые средства решения задач построения, распознавания и порождения;

уметь:

- пользоваться своими знаниями для решения фундаментальных, прикладных и технологических задач в различных предметных областях;
- делать правильные выводы из сопоставления результатов теории и эксперимента, выбирать правильно параметры методов, адекватные размерности обучающих выборок;
- делать качественные и количественные выводы при переходе к предельным условиям в изучаемых проблемах;
- осваивать новые предметные области, теоретические подходы и экспериментальные методики;
- планировать оптимальное проведение обучения по прецедентам;
- эффективно использовать информационные технологии и компьютерную технику для достижения необходимых теоретических и прикладных результатов.

владеть:

- навыками анализа большого объема частично противоречивых и неполных признаков описаний;
- навыками самостоятельной работы в лаборатории с использованием современных компьютерных технологий;
- культурой постановки и планирования последовательности решения задач анализа кода и алгоритмов;
- навыками грамотного анализа кода и алгоритмов задач, оформления результатов анализа и выдачей экспертных оценок по качеству используемого программного кода;
- практикой исследования и решения теоретических и прикладных задач;
- навыками анализа реальных задач из различных предметных областей на уровне отдельных подходов и коллективами алгоритмов;

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Виды учебных занятий, включая самостоятельную работу				
		Лекции	Практические и семинарские занятия	Лаборат. работы	Задания, курсовые работы	Самост. работа
1	Элементы теории алгоритмов. Типы задач. Задачи порождения, вычисления, построения. Задачи распознавания, доказательства, проверки.	2				2
2	Основные объекты формулирования задач. Множества, отношения, алгоритмы, решаемые алгоритмами; алгоритмические исчисления.	2				2

3	Три типа понятия алгоритма: алгоритмические машины; функции, порождаемые алгоритмами; алгоритмические исчисления.	2				2
4	Машины состояний. Блок-схема алгоритма (логическая структура алгоритма – ЛСА) как граф некоторой машины состояний. Регулярные выражения и алгебра регулярных выражений.	2				2
5	Модели программы в виде строки Ляпунова (SL). Эквивалентные преобразования строки Ляпунова. Структурное программирование Дейкстры.	2				2
6	Алгоритмические языки Фортран, Алгол, Паскаль, Си. Сравнительный анализ алгоритмических языков.	6				6
7	Функции, вычисляемые алгоритмом. Конструктор Клини.	2				2
8	Базовое множество и базовый набор функций. Граф подстановок. Информационная структура алгоритма (ИСА), скобочная формула и граф скобочной формулы.	2				2
9	ИСА для рекурсивно-примитивных функций. Граф ИСА из бесконечно повторяющихся фрагментов.	2				2
10	ИСА для рекурсивных функций общего вида. Пример рекурсивной функции Аккермана.	2				4
11	Теорема А.П. Ершова о максимальном размере графа ИСА и о минимальной памяти, необходимой для выполнения программы на одном процессоре.	2				4
12	Функциональные языки программирования: Лисп, Перл, CLU (Б. Лисков), Python, Ruby. Сравнительный анализ функциональных языков.	2				6
13	Функциональные языки программирования: Лисп, Перл, CLU (Б. Лисков), Python, Ruby. Сравнительный анализ функциональных языков.	2				6
14	Разложение графа ИСА для выполнения на n процессорах. Ярусно-параллельная форма (ЯПФ) ИСА.	6				16
15	Логическое программирование и языки логического программирования.	6				16
16	Язык рассуждающих сетей Ван-Хао. Joiner-сети, управляющие рассуждениями.	6				16

17	Асинхронный автомат Малера, моделирующий распространение возбуждающих событий в пространстве Вейча.	6				14
18	Язык Пролог. Хорновские дизъюнкты. Вывод в языке Пролог, использующий метод резолюций.	6				16
Итого часов		60				120
Подготовка к экзамену		0 час.				
Общая трудоёмкость		180 час., 5 зач.ед.				

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 3 (Осенний)

1. Элементы теории алгоритмов. Типы задач. Задачи порождения, вычисления, построения. Задачи распознавания, доказательства, проверки.

Элементы теории алгоритмов. Типы задач. Задачи порождения, вычисления, построения. Задачи распознавания, доказательства, проверки.

2. Основные объекты формулирования задач. Множества, отношения, алгоритмы,аемые алгоритмами; алгоритмические исчисления.

Основные объекты формулирования задач. Множества, отношения, алгоритмы.

3. Три типа понятия алгоритма: алгоритмические машины; функции, порождаемые алгоритмами; алгоритмические исчисления.

Три типа понятия алгоритма: алгоритмические машины; функции, порождаемые алгоритмами; алгоритмические исчисления.

4. Машины состояний. Блок-схема алгоритма (логическая структура алгоритма – ЛСА) как граф некоторой машины состояний. Регулярные выражения и алгебра регулярных выражений.

Машины состояний. Блок-схема алгоритма (логическая структура алгоритма – ЛСА) как граф некоторой машины состояний. Регулярные выражения и алгебра регулярных выражений.

5. Модели программы в виде строки Ляпунова (SL). Эквивалентные преобразования строки Ляпунова. Структурное программирование Дейкстры.

Модели программы в виде строки Ляпунова (SL). Эквивалентные преобразования строки Ляпунова. Структурное программирование Дейкстры.

6. Алгоритмические языки Фортран, Алгол, Паскаль, Си. Сравнительный анализ алгоритмических языков.

Алгоритмические языки Фортран, Алгол, Паскаль, Си. Сравнительный анализ алгоритмических языков.

7. Функции, вычисляемые алгоритмом. Конструктор Клини.

Функции, вычисляемые алгоритмом. Конструктор Клини.

8. Базовое множество и базовый набор функций. Граф подстановок. Информационная структура алгоритма (ИСА), скобочная формула и граф скобочной формулы.

Базовое множество и базовый набор функций. Граф подстановок. Информационная структура алгоритма (ИСА), скобочная формула и граф скобочной формулы.

9. ИСА для рекурсивно-примитивных функций. Граф ИСА из бесконечно повторяющихся фрагментов.

ИСА для рекурсивно-примитивных функций. Граф ИСА из бесконечно повторяющихся фрагментов.

10. ИСА для рекурсивных функций общего вида. Пример рекурсивной функции Аккермана.

ИСА для рекурсивных функций общего вида. Пример рекурсивной функции Аккермана.

11. Теорема А.П. Ершова о максимальном размере графа ИСА и о минимальной памяти, необходимой для выполнения программы на одном процессоре.

Теорема А.П. Ершова о максимальном размере графа ИСА и о минимальной памяти, необходимой для выполнения программы на одном процессоре.

12. Функциональные языки программирования: Лисп, Перл, CLU (Б. Лисков), Python, Rub
Сравнительный анализ функциональных языков.

Функциональные языки программирования: Лисп, Перл, CLU (Б. Лисков), Python, Rub
Сравнительный анализ функциональных языков.

13. Функциональные языки программирования: Лисп, Перл, CLU (Б. Лисков), Python, Rub
Сравнительный анализ функциональных языков.

Семестр: 4 (Весенний)

14. Разложение графа ИСА для выполнения на n процессорах. Ярусно-параллельная форма (ЯПФ) ИСА.

Разложение графа ИСА для выполнения на n процессорах. Ярусно-параллельная форма (ЯПФ) ИСА.

15. Логическое программирование и языки логического программирования.

16. Язык рассуждающих сетей Ван-Хао. Joiner-сети, управляющие рассуждениями.

17. Асинхронный автомат Малера, моделирующий распространение возбуждающих событий в пространстве Вейча.

18. Язык Пролог. Хорновские дизъюнкты. Вывод в языке Пролог, использующий метод резолюций.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Учебная аудитория, оснащенная мультимедиа-проектором и экраном.

6. Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Основная литература

1. Языки программирования. Практический сравнительный анализ [Текст] / М. Бен-Ари ; пер. с англ. В. С. Штаркмана, М. Н. Яковлевой ; под ред. В. С. Штаркмана .— М. : Мир, 2000 .— 366 с.

Дополнительная литература

1. Стили и методы программирования [Текст] : курс лекций : учеб. пособие для вузов / Н. Н. Непейвода .— М. : Интернет-Ун-т Информ. Технологий, 2005 .— 320 с.

7. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

1. Хоггер К. Введение в логическое программирование, - М.: Мир, 1988.
2. Маурер У. Введение в программирование на языке ЛИСП, - М.: Мир, 1976.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

<http://www.intuit.ru>

<http://ru.wikipedia.org>

9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

На лекционных и лабораторных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

10. Методические указания для обучающихся по освоению дисциплины

Студент, изучающий курс сравнительный анализ языков программирования, должен с одной стороны, овладеть общим понятийным аппаратом, а с другой стороны, должен научиться применять теоретические знания на практике.

В результате изучения дисциплины студент должен знать основные определения, понятия, аксиомы, методы доказательств.

Успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой. Самостоятельная работа включает в себя:

- чтение и конспектирование рекомендованной литературы,
- проработку учебного материала (по конспектам лекций, учебной и научной литературе), подготовку ответов на вопросы, предназначенных для самостоятельного изучения, доказательство отдельных утверждений, свойств;
- решение задач, предлагаемых студентам на практических занятиях,
- подготовку к экзамену.

Руководство и контроль за самостоятельной работой студента осуществляется в форме индивидуальных консультаций.

Показателем владения материалом служит знание различных подходов, алгоритмов и методов, а также умение решать задачи. Для формирования умения применять теоретические знания на практике студенту необходимо решать как можно больше задач, в том числе прикладных задач классификации языков программирования. При решении задач каждое действие необходимо аргументировать, ссылаясь на известные теоретические сведения.

Важно добиться понимания изучаемого материала, а не механического его запоминания. При затруднении изучения отдельных тем, вопросов, следует обращаться за консультациями к лектору.

11. Фонд оценочных средств для проведения промежуточной аттестации по итогам обучения

Приложение

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ
ПО ДИСЦИПЛИНЕ**

по направлению: Компьютерная безопасность (специалитет)
профиль подготовки: Безопасность распределенных компьютерных систем
Факультет радиотехники и кибернетики
кафедра информатики и вычислительной математик
курс: 2
квалификация: специалист

Семестры, формы промежуточной аттестации:

3(Осенний) - без промежуточной аттестации

4(Весенний) - Дифференцированный зачет

Разработчик: А.В. Пензин, старший преподаватель

1. Компетенции, формируемые в процессе изучения дисциплины

Освоение дисциплины направлено на формирование у обучающегося следующих общекультурных (ОК), общепрофессиональных (ОПК) и профессиональных (ПК) компетенций:

способность к самоорганизации и самообразованию (ОК-8);

способность к самостоятельному построению алгоритма, проведению его анализа и реализации в современных программных комплексах (ОКП-10);

способность применять методы научных исследований в профессиональной деятельности, в том числе в работе над междисциплинарными и инновационными проектами (ОПК-4);

способность учитывать современные тенденции развития информатики и вычислительной техники, компьютерных технологий в своей профессиональной деятельности, работать с программными средствами общего и специального назначения (ОПК-7).

2. Показатели оценивания компетенций

В результате изучения дисциплины «Сравнительный анализ языков программирования» обучающийся должен:

знать:

- фундаментальные понятия и методы теории алгоритмов;
- современные проблемы анализа алгоритмов, постановки задачи, выбора языка программирования, построения полного решения задачи;
- методы и подходы решения практических алгоритмических задач, от постановки задачи до финального алгоритма;
- инструментальные языковые средства решения задач построения, распознавания и порождения;

уметь:

- пользоваться своими знаниями для решения фундаментальных, прикладных и технологических задач в различных предметных областях;
- делать правильные выводы из сопоставления результатов теории и эксперимента, выбирать правильно параметры методов, адекватные размерности обучающих выборок;
- делать качественные и количественные выводы при переходе к предельным условиям в изучаемых проблемах;
- осваивать новые предметные области, теоретические подходы и экспериментальные методики;
- планировать оптимальное проведение обучения по прецедентам;
- эффективно использовать информационные технологии и компьютерную технику для достижения необходимых теоретических и прикладных результатов.

владеть:

- навыками анализа большого объема частично противоречивых и неполных признаков описаний;
- навыками самостоятельной работы в лаборатории с использованием современных компьютерных технологий;
- культурой постановки и планирования последовательности решения задач анализа кода и алгоритмов;
- навыками грамотного анализа кода и алгоритмов задач, оформления результатов анализа и выдачей экспертных оценок по качеству используемого программного кода;
- практикой исследования и решения теоретических и прикладных задач;
- навыками анализа реальных задач из различных предметных областей на уровне отдельных подходов и коллективами алгоритмов;

3. Перечень типовых контрольных заданий, используемых для оценки знаний, умений, навыков

Промежуточная аттестация по дисциплине «СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ» осуществляется в форме экзамена. Экзамен проводится в устной форме.

Перечень контрольных вопросов:

1. Сформулировать основные варианты типов задач возникающих в теории алгоритмов, привести примеры.
2. Дать определение основных математических механизмов используемых в теории формальных языков.
3. Построение решения реальной задачи от постановки задачи до реализации на конкретном языке.
4. Алгоритм машина, алгоритм функция, алгоритм исчисление.
5. Блок схема алгоритма, алгоритмические машины.
6. Структурное программирование Дейкстры.
7. Регулярные выражения, алгебра регулярных выражений, конструктор Клини.
8. Логическая структура алгоритма, строка Ляпунова, эквивалентные преобразования строки Ляпунова.
9. Парадигмы языков программирования. Призки для сравнения языков программирования.
10. Алгоритмические языки Алгол, Фортран, Паскаль, Си и их сравнительная характеристика.
11. Особенности реализации Алгол-60(ввод-вывод, трюк Йенсена,...). Особенности реализаций языка Фортран.
12. Основные особенности языка Pascal.
13. Языки Си и C++ .
14. Дать сравнительный анализ основных конструкций языков Fortran, Pascal и Си. Привести примеры.
15. Информационная структура алгоритма. Скобочная формула. Граф ИСА.
16. Инвариантные свойства ИСА. Аппаратная реализация ИСА.
17. Грамматики. Конечные и магазинные автоматы. Классификация языков Хомского.
18. Языки Perl, Lisp, CLU, Python и Ruby – сравнительная характеристика.
19. Лямбда исчисление в языках Lisp, Python.
20. Сравнительный анализ основных конструкций функциональных языков. (Perl, Lisp).
21. Особенности реализации объектно-ориентированных конструкций в языках CLU &
22. Ruby.
23. Ярусно-параллельная форма ИСА.
24. Теорема А.П. Ершова о минимальном разрезе графа ИСА.
25. Разложение ИСА на 2-процессорную и n-процессорную систему с общей памятью
26. Распределенное программирование, теорема о тупиках.
27. Сеть Петри.
28. Синхронизация параллельных программ.
29. Языки Ада, Модула, SmallTalk – и их сравнительный анализ.
30. Особенности реализации языка Ada/
31. Логическое программирование. Joiner-сети, асинхронный автомат Малера.
32. Рассуждающие сети Ван-Хао.
33. Язык Пролог и метод резолюций в языке Пролог.

Примеры контрольных заданий:

1. Предложить алгоритм порождения множества целых чисел, которые делятся на три:

- a) Если числа заданы двоичным кодом
 - b) Троичным кодом
 - c) Десятичным кодом
2. Предложить алгоритм кодирования изображения букв русского алфавита, структура которых описывается в виде лабиринта из комнат и коридоров со следующими типами объектов:
 - a) Терминальный объект «комната с единственным коридором»
 - b) Объект «перелом» – «комната с двумя коридорами, идущими под углом»
 - c) Объект «сопряжение» - «комната с n коридорами», где $n > 2$.
3. Предложить блок-схему алгоритма (ЛСА) с тремя вложенными друг в друга циклами. Построить для нее машину состояний и несколько эквивалентных строк Ляпунова (SL). Построение ЛСА выполнить в виде структурированной методики Дейкстры. Показать, что в этом случае SL имеет минимальное число безусловных переходов.
4. Написать регулярное выражение с тремя вложенными операциями «*» (звезда Клини) и построить граф переходов машины состояний, которая порождает все возможные последовательности операторов-действий.
5. В любой программе, написанной на Си выделить логическую структуру (ЛСА), построить для нее граф перехода машины состояний и соответствующее регулярное выражение.

4. Критерии оценивания

Оценка «отлично (10)» выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

Оценка «отлично (9)» выставляется студенту, показавшему систематизированные, глубокие знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

Оценка «отлично (8)» выставляется студенту, показавшему систематизированные, знания учебной программы дисциплины и умение применять их на практике при решении конкретных задач, правильное обоснование принятых решений;

Оценка «хорошо (7)» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, недопускает в ответе или в решении задач некоторые неточности;

Оценка «хорошо (6)» выставляется студенту, если он твердо знает материал, грамотно излагает его, умеет применять полученные знания на практике, недопускает в ответе или в решении задач некоторые неточности;

Оценка «хорошо (5)» выставляется студенту, если он знает материал, грамотно излагает его, умеет применять полученные знания на практике, недопускает в ответе или в решении задач некоторые неточности;

Оценка «удовлетворительно (4)» выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

Оценка «удовлетворительно (3)» выставляется студенту, показавшему фрагментарный характер знаний, недостаточно правильные формулировки базовых понятий, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

Оценка «неудовлетворительно (2)» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Оценка «неудовлетворительно (1)» выставляется студенту, который не знает основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Во время проведения экзамена обучающиеся могут пользоваться программой дисциплины, Интернетом, справочной литературой, вычислительной техникой.

Экзамен проводится путем организации специального опроса, проводимого в устной форме.