

**Федеральное государственное бюджетное учреждение науки  
Институт систем информатики им. А.П. Ершова  
Сибирского отделения Российской академии наук  
(ИСИ СО РАН)**

УТВЕРЖДАЮ  
Директор ИСИ СО РАН



«1» сентября 2015 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Языки и системы программирования.  
Технология разработки программного обеспечения»**

**Направление подготовки:** 09.06.01 «Информатика и вычислительная техника»

**Специальность:** 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»

**Уровень образования:** подготовка кадров высшей квалификации

**Квалификация выпускника:** Исследователь. Преподаватель-исследователь

Составители рабочей программы

д.ф.-м.н. профессор  
(должность, ученое звание, ученая степень)

  
(подпись)

Касьянов В.Н.  
(Ф.И.О.)

Рабочая программа утверждена на заседании Ученого совета Института  
«07» июля 2015 г., протокол № 5-2015

Председатель Ученого совета

  
(подпись)

Марчук А.Г.  
(ФИО)

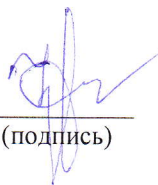
СОГЛАСОВАНО:

Зам. директора

  
(подпись)

Мурзин Ф.А.  
(Ф.И.О.)

Зав. аспирантурой  
и докторантурой

  
(подпись)

Воронко Н.Ф.  
(Ф.И.О.)

## 1. Цели освоения дисциплины

Целью освоения дисциплины «Языки и системы программирования. Технология разработки программного обеспечения» является ознакомление с методами и методологиями разработки программного обеспечения, анализа и проектирования программных систем, освоение инструментальной поддержки процесса разработки программного обеспечения, парадигм и техники программирования.

*(Указываются цели освоения дисциплины)*

## 2. Место дисциплины в структуре основной профессиональной образовательной программы послевузовского профессионального образования (аспирантура)

Данная дисциплина «Языки и системы программирования. Технология разработки программного обеспечения» (Б1.В.ОД.3) относится к группе обязательных дисциплин по специальности 05.13.11.

## 3. Требования к уровню подготовки аспиранта, завершившего изучение данной дисциплины

Аспиранты, завершившие изучение данной дисциплины, должны:

- **знать:** теорию и технологию разработки программного обеспечения
- **уметь:** разрабатывать последовательные и параллельные программы с учетом архитектуры вычислительной системы
- **владеть:** современными средствами разработки программного обеспечения

Компетенции, формируемые у обучающихся, в соответствии с ООП по направлению 09.06.01 «Информатика и вычислительная техника» и профилю (специальности) 05.13.11 «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей»:

<b>Универсальные компетенции:</b>	УК1, УК3, УК5
<b>Общепрофессиональные компетенции:</b>	ОПК1, ОПК3, ОПК4, ОПК6
<b>Профессиональные компетенции:</b>	ПК1, ПК2, ПК3, ПК4, ПК5, ПК6

## 4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 2 зачетных единицы 72 часов.

Вид учебной работы	Объем часов / зачетных единиц
<b>Обязательная аудиторная учебная нагрузка (всего)</b>	22
в том числе:	
лекции	18
семинары	
практические занятия	

контроль самостоятельной работы	4
<b>Самостоятельная работа аспиранта (всего)</b>	50
<b>Вид контроля по дисциплине</b>	экзамен

## 5. Разделы дисциплины и виды занятий

№ п/п	Название раздела дисциплины	Объем часов / зачетных единиц					
		Всего ауд. часов	из них				Самост. работа
			Лекции	семинары	практ. занятия	КСР	
1	Языки программирования	1				0,1	3
2	Языки высокого уровня	1				0,1	3
3	Объектно-ориентированное программирование	1				0,15	4
4	Распределенное программирование	1				0,15	4
5	Основы построения трансляторов	2				0,15	3
6	Анализ исходной программы в компиляторе	2				0,15	4
7	Оптимизация программ при их компиляции	2				0,15	4
8	Генерация объектного кода в компиляторах	2				0,15	4
9	Машинно-зависимые и машинно-ориентированные языки	1				0,15	3
10	Системы программирования (СП), типовые компоненты СП	1				0,15	4
11	Пакеты прикладных программ (ППП)	1				0,15	3
12	Технология разработки и сопровождения программ	1				0,15	4
13	Отладка, тестирование, верификация и оценивание сложности программ	1				0,15	4
14	Методы спецификации программ	1				0,15	3

## 6. Содержание дисциплины:

*(Раздел, тема учебного курса, содержание лекции)*

Языки программирования, их классификация. Трансляторы и их виды. Синтаксис и семантика языков программирования. Цепочки и языки. Порождающие грамматики, распознаватели и преобразователи. Классификация Хомского. Лексемы и понятия, Бэкуса-Наура формы (БНФ) и синтаксические диаграммы. Операционная, трансформационная и денотационная семантика. Стандарты и версии языков программирования.

Языки программирования высокого уровня. Иерархия языковых конструкций. Система типов языка высокого уровня, простые, составные, первичные, стандартные, библиотечные и конструируемые типы. Статическая и динамическая типизация, свойства языков со строгой типизацией. Процедуры и функции, простая структура вызовов-возвратов, задачи, сопрограммы и подпрограммы обработки исключительных ситуаций. Блоки и правила локализации имен. Механизм параметров, ключевые и позиционные параметры, способы подстановки параметров. Рекурсия.

Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).

Распределенное программирование. Процессы и их синхронизация. Семафоры, мониторы Хоара. Объектно-ориентированное распределенное программирование. CORBA. Параллельное программирование над общей памятью. Нити. Стандартный интерфейс Open MP. Распараллеливание последовательных программ. Параллельное программирование над распределенной памятью. Парадигмы SPMD и MIMD. Стандартный интерфейс MPI.

Этапы и схемы трансляции. Построение абстрактной программы: лексический, синтаксический и контекстный анализы. Генерация. Оптимизация и схема оптимизирующей трансляции. Промежуточные представления программ. Обнаружение ошибок.

Конечные автоматы и их свойства. Реализация лексического анализа. Автоматы с магазинной памятью и их свойства. Стратегии синтаксического анализа и обработки ошибок. LL-грамматики и LL-анализаторы. LR-грамматики LR-анализаторы. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик, системы LEX и YACC. Контекстный анализ: идентификация и атрибутивная индукция. Атрибутивные грамматики и алгоритмы вычисления атрибутов. Атрибутивные системы построения трансляторов.

Смешанная стратегия при трансляции и оптимизирующие преобразования. Локальная, квазилокальная и глобальная оптимизация. Схемы программ и промежуточные представления программ. Анализ потока управления: представление и структуризация множеств исполнений, нахождение свойств операторов и переходов, выбор порядка обработки операторов. Анализ потока данных: свойства исполнений и состояний. Способы оптимизации программ: разгрузка участков повторяемости, упрощение действий, чистка программы, экономия памяти, сокращение программы. Выбор последовательности применения оптимизирующих преобразований и контекстных условий.

Общая схема генерации объектного кода в компиляторах и внутреннее представление команд. Представление структур данных. Генерация выражений, операторов и подпрограмм. Распределение памяти. Динамическая поддержка. Покадровая оптимизация. Перенастраиваемые (retargetable) компиляторы. Автоматическая генерация генераторов объектного кода (системы BEG, Iburg и др.).

Машинно-зависимые и машинно-ориентированные языки. Язык ассемблера, представление машинных команд и констант, команды транслятору и их реализация. Макросы и макропроцессоры.

Системы программирования и их типовые компоненты: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы. Применение макросредств при построении трансляторов. Методы обеспечения мобильности. Метод раскрутки. Модульный подход при построении трансляторов.

Пакеты прикладных программ (ППП). Системная часть и наполнение. Языки общения с ППП. Машинная графика. Средства поддержки машинной графики. Графические пакеты.

Технология разработки и сопровождения программ. Жизненный цикл программы. Этапы разработки, степень и пути их автоматизации. Обратная инженерия. Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов. Модули, взаимодействие между модулями, иерархические структуры программ.

Отладка, тестирование, верификация и оценивание сложности программ. Генерация тестов. Системы генерации тестов. Срезы программ (slicing) и их применение при отладке программ и для генерации тестов.

Методы спецификации программ. Методы проверки спецификации. Схемное, структурное, визуальное программирование. Языки функционального и логического программирования. Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.

## **7. Самостоятельная работа аспирантов**

Изучение основной и дополнительной литературы по вопросам программы.  
*(Приводятся виды самостоятельной работы обучающегося, порядок их выполнения и контроля, учебно-методическое обеспечение (возможно в виде ссылок) самостоятельной работы по отдельным видам дисциплин)*

## **8. Учебно-методическое и информационное обеспечение дисциплины**

### **8.1. Основная и дополнительная литература**

а) основная литература:

1. Ахо А., Сети В., Ульман Дж.Д. Компиляторы: принципы, технологии и инструменты. 2-е изд., М.: Вильямс, 2015.
2. Касьянов В.Н. Оптимизирующие преобразования программ. М.: Наука, 1988.
3. Касьянов В.Н., Поттосин И.В. Методы построения трансляторов. Новосибирск: Наука, 1986.
4. Себеста Р.В. Основные концепции языков программирования. 5-е издание. – М.: Вильямс, 2001.
5. Хопкрофт Дж., Мотвани Р., Ульман Дж. Введение в теорию автоматов, языков и вычислений. 2-е изд., исправ., М.: Вильямс, 2016.

б) дополнительная литература

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978. Т.1, 2.
2. Александреску А. Современное проектирование на C++. Обобщенное программирование и прикладные шаблоны проектирования. М.: Вильямс, 2002.

3. Браун П. Макропроцессоры и мобильность программного обеспечения. Серия: Математическое обеспечение ЭВМ. Пер. с англ. М.: Мир, 1977.
4. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. 2-е изд., М.: Бином, 1999.
5. Бутаков Е.А. Методы создания качественного программного обеспечения. М.: Энергоатомиздат, 1984.
6. Ван Тассел Д. Стил, разработка, эффективность, отладка и испытание программ. М.: Мир, 1981.
7. Воеводин В.В. Воеводин Вл.В. Параллельные вычисления. СПб: БХВ-Петербург, 2002.
8. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приёмы объектно-ориентированного проектирования. Паттерны проектирования. СПб: Питер, 2001.
9. Евстигнеев В. А., Касьянов В. Н. Оптимизирующие преобразования в распараллеливающих компиляторах // Программирование. - 1996.- № 6. - С.12-26.
10. Карпов Ю. Г. Model checking. Верификация параллельных и распределенных программных систем. СПб.: БХВ-Петербург, 2010.
11. Касьянов В.Н., Евстигнеев В.А. Графы в программировании: обработка, визуализация и применение. СПб: БХВ-Петербург, 2003.
12. Кларк Э.М., Грамберг О., Пелед Д. Верификация моделей программ. Model Checking. М.: МЦНМО, 2002.
13. Касьянов В. Н. Лекции по теории формальных языков, автоматов и сложности вычислений. Новосибирск: НГУ, 1995.
14. Касьянов В. Н., Мирзуитова И. Л. Slicing: срезы программ и их использование. Новосибирск: ИСИ СО РАН, 2002.
15. Лавров С. Программирование. Математические основы, средства, теория. СПб: БХВ-Петербург, 2001.
16. Майерс Г. Искусство тестирования программ. М.: Финансы и статистика, 1982.
17. Мандел Т. Разработка пользовательского интерфейса. М.: ДМК Пресс, 2001.
18. Скопин И.Н. Понятия и модели жизненного цикла программного обеспечения. Новосибирск: НГУ, 2003.
19. Торрес Р.Дж. Практическое руководство по проектированию и разработке пользовательского интерфейса. М.: Вильямс, 2002.
20. Хендерсон П. Функциональное программирование. М.: Мир, 1983.

в) программное обеспечение и Интернет-ресурсы

## **8.2. Перечень вопросов и заданий (аттестации) и/или тем рефератов**

1. Языки программирования, их классификация.
2. Трансляторы и их виды.
3. Синтаксис и семантика языков программирования.
4. Цепочки и языки.
5. Порождающие грамматики, распознаватели и преобразователи.
6. Классификация Хомского.
7. Лексемы и понятия, Бэкуса-Наура формы (БНФ) и синтаксические диаграммы.

8. Операционная, трансформационная и денотационная семантика.
9. Стандарты и версии языков программирования.
10. Языки программирования высокого уровня. Иерархия языковых конструкций.
11. Система типов языка высокого уровня, простые, составные, первичные, стандартные, библиотечные и конструируемые типы.
12. Статическая и динамическая типизация, свойства языков со строгой типизацией.
13. Процедуры и функции, простая структура вызовов-возвратов, задачи, сопрограммы и подпрограммы обработки исключительных ситуаций.
14. Блоки и правила локализации имен.
15. Механизм параметров, ключевые и позиционные параметры, способы подстановки параметров.
16. Рекурсия.
17. Объектно-ориентированное программирование. Классы и объекты, наследование, интерфейсы.
18. Понятие об объектном окружении. Рефлексия. Библиотеки классов. Средства обработки объектов (контейнеры и итераторы).
19. Распределенное программирование, процессы и их синхронизация, семафоры, мониторы Хоара.
20. Объектно-ориентированное распределенное программирование. CORBA.
21. Параллельное программирование над общей памятью. Нити. Стандартный интерфейс Open MP.
22. Распараллеливание последовательных программ.
23. Параллельное программирование над распределенной памятью. Парадигмы SPMD и MIMD. Стандартный интерфейс MPI.
24. Этапы и схемы трансляции.
25. Построение абстрактной программы: лексический, синтаксический и контекстный анализы. Генерация.
26. Оптимизация и схема оптимизирующей трансляции.
27. Промежуточные представления программ.
28. Обнаружение ошибок при трансляции.
29. Конечные автоматы и их свойства.
30. Реализация лексического анализа.
31. Автоматы с магазинной памятью и их свойства.
32. Стратегии синтаксического анализа и обработки ошибок.
33. LL-грамматики и LL-анализаторы.
34. LR-грамматики LR-анализаторы.
35. Автоматическое построение лексических и синтаксических анализаторов по формальным описаниям грамматик, системы LEX и YACC.
36. Контекстный анализ: идентификация и атрибутивная индукция.
37. Атрибутные грамматики и алгоритмы вычисления атрибутов.
38. Атрибутные системы построения трансляторов.
39. Смешанная стратегия при трансляции и оптимизирующие преобразования. Локальная, квазилокальная и глобальная оптимизация.
40. Схемы программ и промежуточные представления программ.



41. Анализ потока управления: представление и структуризация множеств исполнений, нахождение свойств операторов и переходов, выбор порядка обработки операторов.
42. Анализ потока данных: свойства исполнений и состояний.
43. Способы оптимизации программ: разгрузка участков повторяемости, упрощение действий, чистка программы, экономия памяти, сокращение программы.
44. Выбор последовательности применения оптимизирующих преобразований и контекстных условий.
45. Общая схема генерации объектного кода в компиляторах и внутреннее представления команд.
46. Представление структур данных. Генерация выражений, операторов и подпрограмм.
47. Распределение памяти. Динамическая поддержка.
48. Покадровая оптимизация.
49. Перенастраиваемые (retargetable) компиляторы.
50. Автоматическая генерация генераторов объектного кода (системы BEG, Iburg и др.).
51. Машинно-зависимые и машинно-ориентированные языки.
52. Язык ассемблера, представление машинных команд и констант, команды транслятору и их реализация.
53. Макросы и макропроцессоры.
54. Системы программирования и их типовые компоненты: языки, трансляторы, редакторы связей, отладчики, текстовые редакторы.
55. Применение макросредств при построении трансляторов. Методы обеспечения мобильности.
56. Метод раскрутки.
57. Модульный подход при построении трансляторов.
58. Пакеты прикладных программ (ППП). Системная часть и наполнение. Языки общения с ППП.
59. Машинная графика. Средства поддержки машинной графики. Графические пакеты.
60. Технология разработки и сопровождения программ. Жизненный цикл программы. Этапы разработки, степень и пути их автоматизации.
61. Обратная инженерия.
62. Декомпозиционные и сборочные технологии, механизмы наследования, инкапсуляции, задания типов.
63. Модули, взаимодействие между модулями, иерархические структуры программ.
64. Отладка, тестирование, верификация и оценивание сложности программ.
65. Генерация тестов. Системы генерации тестов.
66. Срезы программ (slicing) и их применение при отладке программ и для генерации тестов.
67. Методы спецификации программ. Методы проверки спецификации.
68. Схемное, структурное, визуальное программирование.
69. Языки функционального и логического программирования.

70. Разработка пользовательского интерфейса, стандарт CUA, мультимедийные среды интерфейсного взаимодействия.

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

Для лекций используется класс, оснащённый мультимедийным проектором и имеющий в составе программное обеспечение MS Office и Acrobat Reader. Литература из основного и вспомогательного списков доступна в электронно-библиотечной системе ИСИ СО РАН и в Мемориальной библиотеке А.П. Ершова (каб. 265). Для контроля самостоятельной работы используется компьютер в классе и персональный компьютер лектора.

---

*(Указывается материально-техническое обеспечение данной дисциплины)*

## ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ В РАБОЧЕЙ ПРОГРАММЕ

за 2015 / 2016 учебный год

В рабочую программу Языки и системы программирования. Технология разработки программного обеспечения

(наименование дисциплины)

Для специальности (тей) 05.13.11

(номер специальности)

Вносятся следующие дополнения и изменения:

Дополнения и изменения внес \_\_\_\_\_

(должность, ФИО, подпись)

Рабочая программа пересмотрена и одобрена на заседании Ученого совета Института

Председатель Ученого совета \_\_\_\_\_

(подпись)

(ФИО)