**Санкт-Петербургский государственный университет**

**Р А Б О Ч А Я П Р О Г Р А М М А**

**УЧЕБНОЙ ДИСЦИПЛИНЫ**

Трансляция языков программирования

Programming Language Translation

**Язык(и) обучения**

русский

Трудоемкость в зачетных единицах: 5

Регистрационный номер рабочей программы: 003667

**Раздел 1. Характеристики учебных занятий**

**1.1. Цели и задачи учебных занятий**

Трансляторы занимают одно из важнейших мест в программном обеспечении, поскольку именно они делают программу, написанную на некотором языке программирования, понятной компьютеру. Несмотря на то, что лишь немногие из слушателей курса будут заниматься разработкой или сопровождением трансляторов, идеи и технологии, излагаемые в курсе, они смогут успешно применить и при разработке другого программного обеспечения, например, текстовых редакторов, систем поиска информации, типографских систем и т. п.

Отдельные параметры курса могут варьироваться по степени сложности в зависимости от уровня подготовки обучающихся. Курс должен быть построен на принципах компетентного, целостного, деятельностного подхода к разработке трансляторов, что предполагает распределение содержания обучения по следующим видам деятельности: изучение основных концепций трансляции, разбиение процесса трансляции на последовательно выполняемые задачи, изучение методов решения каждой из задач.

Основным методологическим принципом построения программы курса, равно как и всей концепции изучения трансляции, в целом, является принцип поэтапного системного накопления знаний и формирования необходимых компетенций по модели: от простого и/или знакомого – к сложному и/или незнакомому, а основной методологической стратегией прохождения отдельных разделов программы является ступенчатость и цикличность, предусматривающие постепенный возврат к ранее усвоенному материалу на более высоком языковом и концептуальном уровне. Иными словами, у обучающегося, первоначально воспринимающего транслятор, как некий черный ящик, постепенно формируется необходимый набор понятий и методов, позволяющий успешно решать задачи, связанные не только с собственно разработкой компиляторов, но с целым классом задач, возникающих при разработке текстовых процессоров или при распознавании образов.

Цель изучения дисциплины: обучение понятиям и методам, сложившимся в течение пятидесятилетней теории и практики разработки трансляторов, развитие у обучающихся доказательного, логического мышления; знакомство с алгоритмами, применяемыми при разработке и реализации трансляторов; знакомство с существующими инструментами, позволяющими автоматизировать процесс решения возникающих задач; подготовка к самостоятельному решению различных алгоритмических задач, принадлежащих к данной предметной области.

**1.2. Требования подготовленности обучающегося к освоению содержания учебных занятий (пререквизиты)**

Программа курса предназначена для обучающихся 3 курса и рассчитана на обучающихся, изучавших основы программирования в объеме двух лет обучения и, таким образом, владеющих навыками в использования трансляторов тех или иных языков программирования.

Максимальная эффективность программы будет обеспечена при условии, что обучающийся:

* владеет определенными знаниями математики в рамках курсов, читаемых на математико-механическом факультете
* владеет основами программирования, достаточными для составления программ
* владеет знаниями в рамках курсов: «Дискретная математика», «Архитектура вычислительных систем», «Алгоритмы и структуры данных», «Математическая логика», «Теория автоматов и формальных языков».

**1.3. Перечень результатов обучения (learning outcomes)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование категории (группы) компетенций | Код и наименование компетенции | Планируемые результаты обучения, обеспечивающие формирование компетенции | Код индикатора и индикатор достижения универсальной компетенции |
|  | 1 | 2 | 3 | 4 |
| 1 | Общепрофессиональные дисциплины | ОПК-1 – способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности | Знание основных понятий теории трансляции, умение применять их на практике | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения |

**1.4. Перечень и объём активных и интерактивных форм учебных занятий**

Аудиторная учебная работа: лекции в объеме 32 часов, практические занятия в объёме 32 часов. Из них в активных и интерактивных формах – 10 часов.

**Раздел 2. Организация, структура и содержание учебных занятий**

**2.1. Организация учебных занятий**

**2.1.1 Основной курс**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Трудоёмкость, объёмы учебной работы и наполняемость групп обучающихся | | | | | | | | | | | | | | | | | | |
| Код модуля в составе дисциплины,  практики и т.п. | Контактная работа обучающихся с преподавателем | | | | | | | | | | | | Самостоятельная работа | | | | Объём активных и интерактивных  форм учебных занятий | Трудоёмкость |
| лекции | семинары | консультации | практические  занятия | лабораторные работы | контрольные работы | коллоквиумы | текущий контроль | промежуточная  аттестация | итоговая аттестация | под руководством преподавателя | в присутствии  преподавателя | сам. раб. с использованием  методических материалов | текущий контроль (сам.раб.) | промежуточная аттестация (сам.раб.) | итоговая аттестация  (сам.раб.) |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | | | | | | | | | | | | | |
| Форма обучения: очная | | | | | | | | | | | | | | | | | | |
| Семестр 5 | 32 |  | 2 | 32 |  |  |  |  | 2 |  |  |  | 84 |  | 28 |  | 10 | 5 |
|  | 2-42 |  | 2-25 | 2-25 |  |  |  |  | 2-25 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 32 |  | 2 | 32 |  |  |  |  | 2 |  |  |  | 84 |  | 28 |  |  | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Виды, формы и сроки текущего контроля успеваемости и промежуточной аттестации | | | | | | |
| Код модуля в составе дисциплины, практики и т.п. | Формы текущего контроля успеваемости | | Виды промежуточной аттестации | | Виды итоговой аттестации  (только для программ итоговой аттестации и дополнительных образовательных программ) | |
| Формы | Сроки | Виды | Сроки | Виды | Сроки |
| ОСНОВНАЯ ТРАЕКТОРИЯ | | | | | | |
| Форма обучения: очная | | | | | | |
| Семестр 5 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Период обучения (модуль): семестр 5.   |  |  |  |  | | --- | --- | --- | --- | | № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов | | 1. | Основные понятия   * Основы компиляции * Проекции | лекции | 6 | | практические занятия | 6 | | по методическим материалам | 16 | | 2. | Front-end   * Лексический анализ * Синтаксический анализ * Семантика языка программирования | лекции | 8 | | практические занятия | 8 | | по методическим материалам | 22 | | 3. | Оптимизации   * Оптимизации | лекции | 2 | | практические занятия | 2 | | по методическим материалам | 8 | | 4. | Back-end   * Генерация кода * Управление памятью * Распределение регистров * Оптимизации, применяемые к ПЯ среднего уровня * Bottom-Up Rewrite System | лекции | 16 | | практические занятия | 16 | | по методическим материалам | 38 | | 5. | Промежуточная аттестация | консультации | 2 | | экзамен | 2 | | самостоятельная работа | 28 |  |  | | --- | | **Модуль 1** «Основные понятия»  **Глава 1**. Основы компиляции  Основные понятия: интерпретатор, компилятор. Различные типы целевых программ. Статическое и динамическое связывание. Т-диаграммы, композиция Т-диаграмм. Способы создания компилятора: прямой, раскрутка, кросс-транслятор. Виртуальные машины.  Императивные и декларативные языки программирования. Синтаксис, семантика и прагматика.  Фазы компиляции. Front-end и back-end. Просмотры. Метод заплат.  **Глава 2**. Проекции.  Проекция – отображение конструкции исходного языка в конструкции целевого языка.  Представление данных (примитивные данные, структуры, массивы, строки, множества, указатели, рекурсивные типы).  Процедуры. Передача параметров. Полиморфные процедуры. Сопрограммы.  Объекты. Наследование. | | **Модуль 2** «Front-end»  **Глава 3.** Лексический анализ.  Конечные автоматы и регулярные выражения. Использование конечных автоматов для реализации лексического анализатора. Инструменты для автоматического построения лексического анализатора.  **Глава 4.** Синтаксический анализ  Способы описания синтаксиса языков программирования: форма Бэкуса-Наура, грамматика ван Вейнгардена. Нисходящие и восходящие анализаторы.  Нисходящие анализаторы. Метод рекурсивного спуска. Преобразование грамматики в LL форму. Использование ANTLR для автоматического построения синтаксического анализатора LL грамматики.  Восходящие анализаторы. Метод перенос-свертка. Управляющая программа. Построение магазинного автомата. Конфликты, возникающие при восходящем анализе, и способы их разрешения. Использование YACC для автоматического построения синтаксического анализатора LR грамматики.  Продолжение анализа после ошибок.  **Глава 5.** Семантика языка программирования  Операционная семантика, аксиоматическая семантика, денотационная семантика.  Идентификация идентификаторов. Область действия. Область видимости. Представление в компиляторе.  Представление типов. Операции над типами. Явные и неявные преобразования типов.  **Модуль 3** «Оптимизации»  **Глава 6.** Оптимизации  Виды оптимизаций: по уровню и по размеру фрагмента. Стадии оптимизации. Зависимость между оптимизациями. Порядок оптимизаций.  Основная техника, используемая при оптимизации: анализ потока управления, анализ потока данных, анализ зависимостей и граф зависимостей.  Граф потока управления, линейные участки, def-use цепочки. Удаление мертвого кода.  Предшественники и преемники. Отношение доминирования. Поиск доминаторов. Доминаторное дерево.  Определения цикла. Вложенные циклы. Инварианты цикла. Чистка цикла вверх и вниз. Развертка циклов.  Анализ времени жизни переменных.  SSA (Static Single Assignment) форма. Построение SSA-формы. Восстановление потокового графа из SSA-формы. Алгоритмы оптимизации, использующие SSA-форму. Удаление мертвого кода. Простое распространение констант. Условное распространение констант. Оптимизации, которые можно произвести вместе с распространением констант: распространение копий, константные вычисления, константные условия, удаление недостижимого кода.  **Модуль 4** «Back-end»  **Глава 7.** Генерация кода.  Задачи, решаемые при генерации кода. Простейший алгоритм (Labeling). Образцы. Покрытие образцами. Сравнение RISC и CICS архитектур.  **Глава 8.** Управление памятью  Явное и неявное управление памятью в языках программирования. Фазы управления памятью. Сборка мусора. Алгоритмы: Reference counts, Ленивый алгоритм Weizenbaum’a,  Mark-and-sweep, Copying collection, Cheney’s algorihm. Учет поколений объектов.  **Глава 9.** Распределение регистров  Граф интерференции и его раскраска.  **Глава 10.** Оптимизации, применяемые к ПЯ среднего уровня.  Локальная и глобальная оптимизация общих подвыражений. Нумерация значений. Изменение инварианта цикла.  **Глава 11.** Bottom-Up Rewrite System  Деревянные языки, деревянные грамматики. Выбор инструкций с использованием древовидных грамматик. | |
|  |
|  |
|  |

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Для освоения дисциплины обучающиеся должны посещать лекции и практические занятия, выполнять задания преподавателей.

**3.1.2 Методическое обеспечение самостоятельной работы**

При самостоятельном изучении теоретического материала и выполнении практических заданий целесообразно использовать ресурсы сети Интернет, руководствуясь программой дисциплины. Может быть полезна книга А. Ахо, Р. Сети, Д. Ульман. Компиляторы. Принципы, технологии и инструменты. 2015, если удастся её раздобыть.

**3.1.3 Методика проведения текущего контроля успеваемости и промежуточной аттестации и критерии оценивания**

Экзамен проводится в устной форме. Билет состоит из двух вопросов, на подготовку ответа на которые даётся не менее одного академического часа (при подготовке можно пользоваться литературой). После ответа на вопросы билета преподаватель вправе задать дополнительные вопросы по любой теме из списка вопросов, вынесенных на экзамен. Количество и содержание дополнительных вопросов – на усмотрение преподавателя, принимающего экзамен. Каждый ответ оценивается по шкале от 0 (нет ответа) до 5 (очень хороший ответ), результирующий процент освоения дисциплины вычисляется следующим образом:

1. Оценки за ответы на два основных вопроса усредняются.
2. Оценки за ответы на дополнительные вопросы усредняются, результат усреднения складывается с оценкой, полученной в п.1.
3. Результат умножается на 10.

Далее применяется следующее правило выставления оценки:

|  |  |  |
| --- | --- | --- |
| Итоговый процент  выполнения, % | Оценка СПбГУ при  проведении экзамена | Оценка  ECTS |
| 90-100 | отлично | A |
| 80-89 | хорошо | B |
| 70-79 | хорошо | C |
| 61-69 | удовлетворительно | D |
| 50-60 | удовлетворительно | E |
| менее 50 | неудовлетворительно | F |

**3.1.4 Методические материалы для проведения текущего контроля успеваемости и промежуточной аттестации (контрольно-измерительные материалы, оценочные средства)**

Примерный список вопросов к экзамену:

1. Понятия интерпретатора и компилятор. Различные типы целевых программ. Статическое и динамическое связывание. Т-диаграммы, композиция Т-диаграмм.
2. Способы создания компилятора: прямой, раскрутка, кросс-транслятор. Виртуальные машины.
3. Императивные и декларативные языки программирования. Синтаксис, семантика и прагматика.
4. Фазы компиляции. Front-end и back-end. Просмотры. Метод заплат.
5. Проекция.
6. Представление данных (примитивные данные, структуры, массивы, строки, множества, указатели, рекурсивные типы).
7. Процедуры. Передача параметров. Полиморфные процедуры. Сопрограммы.
8. Объекты. Наследование.
9. Конечные автоматы и регулярные выражения.
10. Использование конечных автоматов для реализации лексического анализатора. Инструменты для автоматического построения лексического анализатора.
11. Способы описания синтаксиса языков программирования: форма Бэкуса-Наура, грамматика ван Вейнгардена. Нисходящие и восходящие анализаторы.
12. Нисходящие анализаторы. Метод рекурсивного спуска.
13. Преобразование грамматики в LL форму.
14. Использование ANTLR для автоматического построения синтаксического анализатора LL грамматики.
15. Восходящие анализаторы. Метод перенос-свертка. Управляющая программа.
16. Построение магазинного автомата. Конфликты, возникающие при восходящем анализе, и способы их разрешения.
17. Использование YACC для автоматического построения синтаксического анализатора LR грамматики.
18. Продолжение анализа после ошибок.
19. Операционная семантика, аксиоматическая семантика, денотационная семантика.
20. Идентификация идентификаторов. Область действия. Область видимости. Представление в компиляторе.
21. Представление типов. Операции над типами. Явные и неявные преобразования типов.
22. Виды оптимизаций: по уровню и по размеру фрагмента. Стадии оптимизации. Зависимость между оптимизациями. Порядок оптимизаций.
23. Основная техника, используемая при оптимизации: анализ потока управления, анализ потока данных, анализ зависимостей и граф зависимостей.
24. Граф потока управления, линейные участки, def-use цепочки. Удаление мертвого кода.
25. Предшественники и преемники. Отношение доминирования. Поиск доминаторов. Доминаторное дерево.
26. Определения цикла. Вложенные циклы. Инварианты цикла. Чистка цикла вверх и вниз. Развертка циклов.
27. Анализ времени жизни переменных.
28. SSA (Static Single Assignment) форма. Построение SSA-формы. Восстановление потокового графа из SSA-формы.
29. Алгоритмы оптимизации, использующие SSA-форму. Удаление мертвого кода. Простое распространение констант. Условное распространение констант. Оптимизации, которые можно произвести вместе с распространением констант: распространение копий, константные вычисления, константные условия, удаление недостижимого кода.
30. Задачи, решаемые при генерации кода. Простейший алгоритм (Labeling).
31. Образцы. Покрытие образцами. Сравнение RISC и CICS архитектур.
32. Явное и неявное управление памятью в языках программирования. Фазы управления памятью. Сборка мусора.
33. Алгоритмы: Reference counts, Ленивый алгоритм Weizenbaum’a,
34. Mark-and-sweep, Copying collection, Cheney’s algorihm. Учет поколений объектов.
35. Граф интерференции и его раскраска.
36. Локальная и глобальная оптимизация общих подвыражений. Нумерация значений. Изменение инварианта цикла.
37. Деревянные языки, деревянные грамматики. Выбор инструкций с использованием древовидных грамматик.

|  |  |  |
| --- | --- | --- |
| № | Код индикатора и индикатор достижения универсальной компетенции | Контрольно-измерительные материалы (КИМ) (тестовые вопросы, контрольные задания, кейсы и пр.) |
|  | 1 | 2 |
| 1 | ОПК-1.1 Уметь идентифицировать возможные проблемы и пути их решения | ответы на вопросы экзамена и на дополнительные вопросы независимо оцениваются по шкале от 0 (нет ответа) до 10 (очень хорошо), далее оценка усредняется. Результат переводится в диапазон от 0 до 100 |

**3.1.5 Методические материалы для оценки обучающимися содержания и качества учебного процесса**

Для оценки обучающимися содержания и качества учебного процесса применяется анкетирование в соответствии с методикой и графиком, утвержденными в установленном порядке.

**3.2. Кадровое обеспечение**

**3.2.1 Образование и (или) квалификация штатных преподавателей и иных лиц, допущенных к проведению учебных занятий**

К преподаванию дисциплины могут быть допущены преподаватели, имеющие диплом о высшем образовании по соответствующему направлению.

**3.2.2 Обеспечение учебно-вспомогательным и (или) иным персоналом**

Специальных требований нет.

**3.3.1 Характеристики аудиторий (помещений, мест) для проведения занятий**

Учебные аудитории для проведения учебных занятий, оснащенные стандартным оборудованием, используемым для обучения в СПбГУ в соответствии с требованиями материально-технического обеспечения

**3.3.2 Характеристики аудиторного оборудования, в том числе неспециализированного компьютерного оборудования и программного обеспечения общего пользования**

Стандартное оборудование, используемое для обучения в СПбГУ. MS Windows, MS Office, Mozilla FireFox, Google Chrome, Acrobat Reader DC, WinZip, Антивирус Касперского.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Специализированное программное обеспечение для компьютерных классов: Visual Studio, ANTLR, YACC.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Для аудиторий с маркерными досками необходимы стирающиеся маркеры в объёме, достаточном для проведения курса. Для аудиторий с меловыми досками необходим мел в объёме, достаточном для проведения курса. Канцелярские принадлежности в объёме, достаточном для проведения курса.

**3.4. Информационное обеспечение**

**3.4.1 Список литературы**

Не предусмотрен.

**3.4.2. Иные информационные источники**

Сайт Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/>

Электронный каталог Научной библиотеки им. М. Горького СПбГУ: <http://www.library.spbu.ru/cgi-bin/irbis64r/cgiirbis_64.exe?C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

Перечень электронных ресурсов, находящихся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/>

Перечень ЭБС, на платформах которых представлены российские учебники, находящиеся в доступе СПбГУ: <http://cufts.library.spbu.ru/CRDB/SPBGU/browse?name=rures&resource%20type=8>

**Раздел 4. Разработчики программы**

Литвинов Юрий Викторович, доцент кафедры системного программирования, [y.litvinov@spbu.ru](mailto:y.litvinov@spbu.ru), по материалам Вояковской Натальи Николаевны, Natalia.Voyakovskaya@lanit-tercom.com