**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: 3

Регистрационный номер рабочей программы: 002290

Санкт-Петербург

2020

**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 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 третьего 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 или аналогичных;
* 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» или аналогичных.

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

Основной целью курса является овладение обучающимися понятиями и алгоритмами и применение полученных знаний как при изучении дисциплин, которые будут читаться в дальнейшем, так и в их будущей практической деятельности.

Компетенции, которые формируются у обучающихся в процессе изучения дисциплины:

* ПКП-4 – способен применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях;
* ПКП-5 – способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;
* ПКП-6 – способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности.

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

Занятия в активных формах: лекции, предполагающие дискуссию с преподавателем, 4 академических часа.

**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 7 | 48 |  | 2 |  |  |  |  |  | 2 |  |  |  | 28 |  | 28 |  | 4 | 3 |
|  | 2-100 |  | 2-100 |  |  |  |  |  | 2-100 |  |  |  | 1-1 |  | 1-1 |  |  |  |
| ИТОГО | 48 |  | 2 |  |  |  |  |  | 2 |  |  |  | 28 |  | 28 |  |  | 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 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 7 |  |  | экзамен, устно, традиционная форма | по графику промежуточной аттестации |  |  |

**2.2. Структура и содержание учебных занятий**

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование темы (раздела, части) | Вид учебных занятий | Количество часов |
| 1. | Основные понятия | лекции | 10 |
| по методическим материалам | 4 |
| 2. | Лексический анализ | лекции | 6 |
| по методическим материалам | 5 |
| 3. | Синтаксический анализ | лекции | 6 |
| по методическим материалам | 5 |
| 4. | Управление выполнением и работа с памятью | лекции | 10 |
| по методическим материалам | 5 |
| 5. | Интерпретация и генерация кода | лекции | 10 |
| по методическим материалам | 5 |
| 6. | Оптимизация | лекции | 6 |
| по методическим материалам | 4 |
|  | Промежуточная аттестация | самостоятельная работа | 28 |
| консультации | 2 |
| экзамен | 2 |
| **Итого** | | | **108** |

Раздел 1

* Обзор языков программирования: история языков программирования; краткий обзор парадигм программирования; роль трансляции в процессе программирования
* Основные вопросы проектирования языков программирования: основные принципы разработки языков программирования; цели разработки; виды эквивалентности типов; модели данных; модели конструкций управления; механизмы абстракции
* Виртуальные машины: понятие виртуальной машины; иерархия виртуальных машин; промежуточные языки
* Введение в теорию трансляции: сравнение интерпретаторов и компиляторов; стадии трансляции; машинно-зависимая и машинно-независимая части транслятора; трансляция как одна из задач программной инженерии

Раздел 2

* Лексический анализ: применение регулярных выражений в программах лексического анализа; ручное кодирование и автоматическая генерация лексических анализаторов; формальное определение лексем; реализация конечного автомата

Раздел 3

* Синтаксический анализ: формальное определение грамматик; BNF и EBNF; нисходящий и восходящий анализ; табличные синтаксические анализаторы и метод рекурсивного спуска; управление таблицами символов; использование средств поддержки процесса трансляции

Раздел 4

* Модели управления выполнением: порядок вычисления подвыражений; исключения и их обработка; системы динамической поддержки
* Работа с объявлениями и модульностью; и управление размещением в памяти: виды объявлений; механизмы параметризации; параметризация типов; механизмы разделения и ограничения областей видимости; сборка мусора
* Системы типов: тип данных как набор значений с операциями над ними; типы данных; модели проверки типов; семантические модели типов, определяемых пользователем; параметрический полиморфизм; полиморфизм подтипа; алгоритмы проверки типов

Раздел 5

1. Интерпретация: итеративная и рекурсивная интерпретации; итеративная интерпретация промежуточного представления; рекурсивная интерпретация дерева разбора программы
2. Генерация кода: промежуточное представление и объектный код; промежуточные представления; реализация генераторов кода; генерация кода путем обхода дерева; контекстно-зависимая трансляция; использование регистров

Раздел 7

* Оптимизация: машинно-независимая оптимизация; анализ потоков данных; оптимизации циклов; машинно-зависимая оптимизация.

**Раздел 3. Обеспечение учебных занятий**

**3.1. Методическое обеспечение**

**3.1.1 Методические указания по освоению дисциплины**

Методические материалы включают в себя следующие типы материалов — учебники, учебные пособия, Интернет-ресурсы.

**3.1.2 Методическое обеспечение самостоятельной работы**

Самостоятельная работа обучающегося, как вид деятельности, стимулирующий активность, самостоятельность, познавательный интерес с целью поиска необходимой информации, приобретения знаний, использования этих знаний для решения учебных, научных и профессиональных задач, представляет собой важную составляющую учебного процесса, которой отводится не менее половины учебного времени при очной форме обучения. Время, отводимое на самостоятельную работу, должно использоваться обучающимися для наиболее полного освоения учебной дисциплины. Следовательно, организация эффективной внеаудиторной самостоятельной работы в процессе обучения требует, с одной стороны, создание условий, призванных обеспечить рациональное и планомерное управление учебной деятельностью, протекающей в отсутствие преподавателя, и тщательной подготовки целого ряда учебных пособий, снабженных методическими указаниями, с другой стороны.

К числу методических пособий относятся:

* общие методические рекомендации и указания по самостоятельной работе;
* фонд контрольных заданий и тестов для самоконтроля, которые позволяют оценить уровень знаний, навыков и умений обучающихся согласно требованиям курса, государственным стандартам и европейским компетенциям.

Роль преподавателя в организации самостоятельной работы состоит в координации действий обучающихся в освоении дисциплины, в методическом и организационном обеспечении учебного процесса. Взаимодействие между преподавателем и обучающимся осуществляется в форме консультаций. Преподаватели также оказывают помощь обучающимся по планированию и организации самостоятельной работы.

Контроль за самостоятельной работой может осуществляться в форме коротких опросов и тестов, углубленных вопросов по темам занятий, дополнительных вопросов, и т.д.

Перечень примерных контрольных вопросов и заданий для самостоятельной работы обучающихся по всем модулям:

1. Особенности выбора компилятора или интерпретатора для реализации языка программирования.
2. Построение проекцией конструкций исходного языка программирования в конструкции целевого языка.
3. Особенности выбора нисходящего или восходящего анализатора для выполнения синтаксического анализа входного языка.
4. Роль оптимизационных преобразований. Различные уровни оптимизаций.
5. Особенности выбора подходящего метода генерации кода.

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

*Методика проведения экзамена*

Экзамен проводится в устной форме с предварительной подготовкой. Обучающемуся предлагается 2 вопроса. При необходимости обучающему могут быть заданы несколько дополнительных вопросов по всем изученным темам. Не разрешается пользоваться никакими вспомогательными материалами и устройствами помимо ручки и чистой бумаги. В случае обнаружения факта использования недозволенных материалов (устройств) составляется акт, и обучающийся удаляется с экзамена.

За каждый вопрос обучающийся может получить от 0 (нет ответа) до 10 (очень хороший ответ) баллов. Результирующий процент выполнения целей обучения определяется как среднее полученных за ответы оценок, переведённых в диапазон от 0 до 100:

|  |  |  |
| --- | --- | --- |
| Итоговый процент выполнения, % | Оценка СПбГУ при  проведении зачёта | 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)**

**3.1.4.1. Формируемые дисциплиной компетенции**

Компетенции, развиваемые дисциплиной:

ПКП-4 – способен применять современные информационные технологии при проектировании, реализации, оценке качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях;

ПКП-5 – способен использовать основные методы и средства автоматизации проектирования, реализации, испытаний и оценки качества при создании конкурентоспособного программного продукта и программных комплексов, а также способен использовать методы и средства автоматизации, связанные с сопровождением, администрированием и модернизацией программных продуктов и программных комплексов;

ПКП-6 – способен использовать знания направлений развития компьютеров с традиционной (нетрадиционной) архитектурой; современных системных программных средств: операционных систем, операционных и сетевых оболочек, сервисных программ; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов в профессиональной деятельности

Шкалы оценивания всех компетенций линейные, степень сформированности определяется долей успешно выполненных заданий, проверяющих данную компетенцию

**3.1.4.2. Контрольно-измерительные материалы (примеры)**

*Пример списка вопросов для устного экзамена*

1. Обзор языков программирования: история языков программирования; краткий обзор парадигм программирования; роль трансляции в процессе программирования
2. Основные вопросы проектирования языков программирования: основные принципы разработки языков программирования; цели разработки; виды эквивалентности типов; модели данных; модели конструкций управления; механизмы абстракции
3. Виртуальные машины: понятие виртуальной машины; иерархия виртуальных машин; промежуточные языки
4. Введение в теорию трансляции: сравнение интерпретаторов и компиляторов; стадии трансляции; машинно-зависимая и машинно-независимая части транслятора; трансляция как одна из задач программной инженерии
5. Лексический анализ: применение регулярных выражений в программах лексического анализа; ручное кодирование и автоматическая генерация лексических анализаторов; формальное определение лексем; реализация конечного автомата
6. Синтаксический анализ: формальное определение грамматик; BNF и EBNF; нисходящий и восходящий анализ; табличные синтаксические анализаторы и метод рекурсивного спуска; управление таблицами символов; использование средств поддержки процесса трансляции
7. Модели управления выполнением: порядок вычисления подвыражений; исключения и их обработка; системы динамической поддержки
8. Работа с объявлениями и модульностью; и управление размещением в памяти: виды объявлений; механизмы параметризации; параметризация типов; механизмы разделения и ограничения областей видимости; сборка мусора
9. Системы типов: тип данных как набор значений с операциями над ними; типы данных; модели проверки типов; семантические модели типов, определяемых пользователем; параметрический полиморфизм; полиморфизм подтипа; алгоритмы проверки типов
10. Интерпретация: итеративная и рекурсивная интерпретации; итеративная интерпретация промежуточного представления; рекурсивная интерпретация дерева разбора программы
11. Генерация кода: промежуточное представление и объектный код; промежуточные представления; реализация генераторов кода; генерация кода путем обхода дерева; контекстно-зависимая трансляция; использование регистров
12. Оптимизация: машинно-независимая оптимизация; анализ потоков данных; оптимизации циклов; машинно-зависимая оптимизация.

**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. Материально-техническое обеспечение**

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

В аудиториях, где проводятся занятия, необходимо наличие досок и средств письма на них.

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

Аудитории для проведения практических занятий должны быть оснащены проекционной техникой и компьютером.

**3.3.3 Характеристики специализированного оборудования**

Специальных требований нет.

**3.3.4 Характеристики специализированного программного обеспечения**

Не требуется.

**3.3.5 Перечень и объёмы требуемых расходных материалов**

Фломастеры цветные или мел, губки, канцелярские товары в объеме, необходимом для организации и проведения занятий по заявкам преподавателей, подаваемым в установленные сроки, доступ преподавателя и обучающихся к в компьютерные классы.

**3.4. Информационное обеспечение**

**3.4.1 Список обязательной литературы**

1. А. Ахо, Р. Сети, Д. Ульман. Компиляторы. Принципы, технологии и инструменты. 2001, ISBN 5-8459-0189-8
2. Касьянов В.Н. Оптимизирующие преобразования программ --М.: Наука. Гл. ред. физ.-мат. лит., 1988.
3. Andrew W. Appel. Modern Compiler Implementation in C. Published by Cambridge University Press, ISBN 0-521-60765-5
4. Andrew W. Appel. Modern Compiler Implementation in ML. Published by Cambridge University Press, ISBN 0-521-60764-7

**3.4.2 Список дополнительной литературы**

1. Richard Jones, Rafael Lins. Garbage Collection. John Wiley & Sons. New York, ISBN 0-471-94148-4.
2. Christopher Fraser, David Hanson. A retargetable C compiler: design and implementation. Addison-Wesley Publishing Company, California, ISBN 0-8053-1670-1.

**3.4.3 Перечень иных информационных источников**

Ресурсы сети Интернет.

**Раздел 4. Разработчики программы**

Терехов Андрей Николаевич, д.ф-м.н, профессор, +7 (812) 4284233,   
a.terekhov@spbu.ru