

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Томский государственный университет

Факультет прикладной математики и кибернетики

УТВЕРЖДАЮ
Декан факультета
прикладной
математики и
кибернетики,
профессор
_____А.М. Горцев

"28" августа 2014 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Автоматное программирование»**

Наименование магистерской программы:
Математическое и программное обеспечение вычислительных машин и компьютерных сетей

Направление подготовки:
01.04.02 – Прикладная математика и информатика

Квалификация (степень) выпускника:
Магистр

Томск
2014г.

1. Цели освоения дисциплины:

В курсе «Автоматное программирование» рассматриваются основные идеи и понятия, описываются математические основы автоматного программирования. Выполняется обзор традиционных областей применения конечных автоматов. Использование автоматного программирования целесообразно для целого ряда задач. Автоматные модели, используемые при создании различных видов программных систем, могут отличаться друг от друга. По этой причине, отдельно описываются различия автоматных моделей. Также рассматривается процедурное программирование с явным выделением состояний и объектно-ориентированное программирование с явным выделением состояний.

2. Место дисциплины в структуре магистерской программы

Дисциплина по выбору студента, читается в 12 семестре (профессиональный цикл, вариативная часть). За курс отвечает кафедра программирования.

Для успешного освоения дисциплины студент должен иметь предварительную подготовку по дисциплинам: дискретная математика, дополнительные главы дискретной математики, языки программирования и методы трансляции, методы компиляции.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины

ОК-6 Способность совершенствовать и развивать свой интеллектуальный и общекультурный уровень, добиваться нравственного и физического совершенствования своей личности; ОК-7 способность и готовность к активному общению в научной, производственной и социально-общественной сферах деятельности.

ПК-2 Способность разрабатывать концептуальные и теоретические модели решаемых научных проблем и задач; ПК-3 Способность углубленного анализа проблем, постановки и обоснования задач научной и проектно-технологической деятельности; ПК-12 способность участвовать в деятельности профессиональных сетевых сообществ по конкретным направлениям.

В результате освоения дисциплины обучающийся должен:

- Знать: основные принципы парадигмы автоматного программирования, типы автоматных моделей, основные методики реализации конечных автоматов.
- Уметь: строить специализированные автоматные модели, а также реализовывать построенные автоматные модели на языках программирования C++, C#.
- Владеть: навыками использования автоматного подхода при проектировании программного обеспечения.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зачетных единицы 108 часов.

4.1. Распределение часов курса по темам и видам работ

№№ п/п	Раздел дисциплины	Всего часов	Аудиторные занятия (час)	Самостоя тельная работа
			в том числе	

			Лекции	Семинары	Лабораторные работы	
1.	Основные понятия автоматного программирования	3	1			2
2.	Автоматные модели	6	2			4
3.	Логическое управление	6	2			4
4.	Процедурное программирование с явным выделением состояний	25	5		12	8
5.	Объектно-ориентированное программирование с явным выделением состояний	34	8		18	8
6.	Спецификации	7	1			6
7.	Инструментальные средства автоматного программирования	7	1			6
8.	Верификация автоматных программ	8	4			4
9.	Примеры Классы задач	10	4			6
10.	Проверка знаний	2	2			
Итого		108	30		30	48

4.2. Перечень разделов курса

Автоматное программирование — это парадигма программирования, при использовании которой программа или её фрагмент осмысливается как модель какого-либо формального автомата.

В зависимости от конкретной задачи в автоматном программировании могут использоваться как конечные автоматы, так и автоматы более сложной структуры.

Определяющими для автоматного программирования являются следующие особенности:

1. Временной период выполнения программы разбивается на шаги автомата, каждый из которых представляет собой выполнение определённой (одной и той же для каждого шага) секции кода с единственной точкой входа; такая секция может быть оформлена, например, в виде отдельной функции и может быть разделена на подсекции, соответствующие отдельным состояниям или категориям состояний.

2. Передача информации между шагами автомата осуществляется только через явно обозначенное множество переменных, называемых состоянием автомата; между шагами автомата программа (или её часть, оформленная в автоматном стиле) не может содержать неявных элементов состояния, таких как значения локальных переменных в стеке, адреса возврата из функций, значение текущего счётчика команд и т.п.; иначе говоря, состояние программы на любые два момента входа в шаг автомата могут различаться между собой только значениями переменных, составляющих состояние автомата (причём такие переменные должны быть явно обозначены в качестве таковых).

Полностью выполнение кода в автоматном стиле представляет собой цикл (возможно, неявный) шагов автомата.

Название автоматное программирование оправдывается ещё и тем, что стиль мышления (восприятия процесса исполнения) при программировании в этой технике практически точно воспроизводит стиль мышления при составлении формальных автоматов (таких как машина Тьюринга, автомат Маркова и др.).

Автоматное программирование широко применяется при построении лексических анализаторов (классические конечные автоматы) и синтаксических анализаторов (автоматы с магазинной памятью).

Кроме того, мышление в терминах конечных автоматов (то есть разбиение исполнения программы на шаги автомата и передача информации от шага к шагу через состояние) необходимо при построении событийно-ориентированных приложений. В этом случае программирование в стиле конечных автоматов оказывается единственной альтернативой порождению множества процессов или потоков управления.

Часто понятие состояний и машин состояний используется для спецификации программ. Так, при проектировании программного обеспечения с помощью UML для описания поведения объектов используются диаграммы состояний (state machine diagrams). Кроме того, явное выделение состояний используется в описании сетевых протоколов.

В качестве одного из примеров применения автоматной парадигмы можно назвать систему STAT ; эта система, в частности, включает встроенный язык STATL, имеющий чисто автоматную семантику.

Существуют также предложения по использованию автоматного программирования в качестве универсального подхода к созданию компьютерных программ вне зависимости от предметной области.

Разделы курса:

1. Основные понятия автоматного программирования. Классы программных систем. Парадигма автоматного программирования. Сущности со сложным поведением. Объект управления. Управляющий автомат.
2. Автоматные модели. Конечные автоматы. Абстрактные автоматы. Структурные автоматы. Автоматизированные объекты управления. Типы автоматизированных объектов управления. Построение автоматных моделей.
3. Логическое управление. Задачи логического управления. Построения автоматных моделей для задач логического управления.
4. Процедурное программирование с явным выделением состояний. Программные системы, управляемые одним автоматом. Программные системы, управляемые множеством взаимодействующих автоматов. Архитектура системы, управляемой множеством автоматов. Вложенные автоматы. Механизмы взаимодействия между автоматами. Автоматная декомпозиция. Критерии автоматной декомпозиции.
5. Объектно-ориентированное программирование с явным выделением состояний. Автоматизированные объекты управления как классы. Объектная декомпозиция. Автоматизированные классы. Методы реализации автоматизированных классов.
6. Спецификации. Спецификация структуры. Спецификация поведения. Схемы связей автомата. Диаграммы взаимодействия. Диаграммы потока данных. Использование UML.
7. Инструментальные средства автоматного программирования. Использование инструментальных средств разработки. MetoAut. UniMod. Описанные на языке XSLT. Шаблоны для языков C++, C#.
8. Верификация автоматных программ. Построение модели Крипки по автоматной модели. Метод автоматных переходов. Метод установки состояний на событиях и входных воздействиях. Метод полного графа

переходов. Метод редуцированного графа переходов. Преобразование сценария для модели Крипки в сценарий для автомата.

9. Классы задач. Генетические алгоритмы. Документооборот. Задачи дискретной математики. Проектирование графического интерфейса. Конечные автоматы в JavaScript

4.3. Лабораторный практикум/практикум на ЭВМ

В курсе предусмотрено выполнение двух лабораторных работ по темам «Процедурное программирование с явным выделением состояний» и «Объектно-ориентированное программирование с явным выделением состояний».

4.4. Практические занятия

Практические занятия не предусмотрены.

4.5. Курсовой проект (курсовая работа)

Курсовой проект не предусмотрен.

5. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Все необходимое учебно-методическое обеспечение по дисциплине представлено в печатном или электронном виде в библиотеке ТГУ, а также в электронном виде в сети Интернет на сайте кафедры программирования или ИДО ТГУ.

В качестве промежуточного контроля успеваемости в середине семестра (на 9 неделе) проводится письменное тестирование.

В конце 12 семестра сдается экзамен.

6. Учебно-методическое и информационное обеспечение дисциплины

а) Основная литература:

1. Шалыто А.А. Автоматное программирование / Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению "Информационно-телекоммуникационные системы", 2008. - 44 с.
2. Шалыто А.А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб.: Наука, 1998. 628 с. УДК 681.3.06:62-507.
3. Шалыто А.А. Логическое управление. Методы аппаратной и программной реализации СПб.: Наука, 2000., 780 с. УДК 681.3.06: 62-507
4. Wagner, F., "Modeling Software with Finite State Machines: A Practical Approach", Auerbach Publications, 2006, ISBN 0-8493-8086-3.

б) Дополнительная литература:

1. Непейвода Н.И. Стили и Методы программирования. М.: Интернет-университет информационных технологий, 2005. 316 с.
2. Ослэндер Д., Риджли Д., Ринггенберг Д. Управляющие программы для механических систем. Объектно-ориентированное проектирование систем реального времени. М.: БИНОМ. Лаборатория знаний. 2004.
3. Канжелев С. Ю., Шалыто А. А. Автоматическая генерация автоматного кода // Информационно-управляющие системы. 2006. № 6.
4. Pring E. J. Finite state machines in JavaScript(<http://www.ibm.com>)

в) Перечень иных информационных источников:

7. Материально-техническое обеспечение дисциплины

7.1. Требования к аудиториям (помещениям, местам) для проведения занятий

Стандартно оборудованные лекционные аудитории. Для проведения лабораторных занятий – компьютерные классы. А также аудитории для проведения интерактивных лекций: видеопроектор, экран настенный, др. оборудование.

7.2. Требования к специализированному оборудованию

Рабочие места преподавателя и студентов должны быть оснащены оборудованием не ниже: Pentium III-800/ОЗУ-256 Мб / Video-32 Мб / HDD 80 Гб / CD-ROM – 48x / Network adapter – 10/100/ Мбс / SVGA – 15”.

7.3. Требования к специализированному программному обеспечению

При использовании электронных учебных пособий каждый обучающийся во время занятий и самостоятельной подготовки должен быть обеспечен рабочим местом в компьютерном классе с выходом в Интернет и корпоративную сеть факультета. Лаборатории (компьютерные классы) должны быть обеспечены необходимым комплектом лицензионного программного обеспечения системы Visual Studio C++.

Программа составлена в соответствии с требованиями ФГОС ВПО с учетом рекомендаций и ПрООП ВПО по направлению подготовки 01.04.02 – Прикладная математика и информатика

Автор: к.т.н., доцент С.А. Останин, ст. преподаватель Е.А. Николаева.

Рецензент: д.т.н., профессор А.Ю. Матросова.

Программа одобрена на заседании Ученого Совета ФПМК от 28.08.2014 г., протокол № 315.