

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ» (ТУСУР)
Кафедра автоматизации обработки информации

ПРАКТИКУМ НЕПРОЦЕДУРНОГО ПРОГРАММИРОВАНИЯ

Методические указания к практическим занятиям и организации
самостоятельной работы
для студентов направления «Программная инженерия»
(уровень бакалавриата)

2018

Салмина Нина Юрьевна

Практикум непроцедурного программирования: Методические указания к практическим занятиям и организации самостоятельной работы для студентов направления «Программная инженерия» (уровень бакалавриата) / Н.Ю. Салмина. – Томск, 2018. – 20 с.

В настоящих методических указаниях изложены рекомендации по подготовке к практическим занятиям, определено задание на каждое занятие и варианты для выполнения индивидуальных работ. Рассмотрены также методические указания для организации самостоятельной работы при изучении дисциплины «Практикум непроцедурного программирования».

Для студентов высших учебных заведений, обучающихся по направлению «Программная инженерия».

Оглавление

1 Введение.....	4
2 Методические указания к проведению практических занятий.....	5
2.1 Практическое занятие «Исследование предметной области»	5
2.2 Практическое занятие «Определение системы»	12
2.3 Практическое занятие «Проектирование системы»	13
2.4 Практическое занятие «Разработка системы»	14
2.4 Практическое занятие «Тестирование системы»	15
3 Методические указания для организации самостоятельной работы	16
3.1 Общие положения	16
3.2 Самостоятельное изучение тем теоретической части курса	16
3.2.1 Модели представления знаний: фреймы	16
3.2.2 Модели представления знаний: семантические сети	17
3.2.3 Обработка и хранение знаний: свойства символов, ассоциативные списки	17
3.2.4 Использование технологий функционального программирования для создания интеллектуальных систем	18
3.2.5 Тестирование программного обеспечения.....	18
4 Рекомендуемая литература	20

1 Введение

Целью дисциплины является формирование у студентов профессиональных знаний и практических навыков по разработке и созданию моделей интеллектуальных систем с помощью языков функционального и логического программирования

Задачи изучения дисциплины:

Получить практические навыки использования методов и инструментальных средств исследования объектов профессиональной деятельности.

Овладеть навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения с использованием языков функционального и логического программирования.

В процессе выполнения практических заданий студент должен продемонстрировать умение использовать современные информационные технологии методов сбора, представления, хранения, обработки и передачи информации с использованием языков функционального и логического программирования. Студент должен получить навыки создания, отладки и тестирования программ, представления результатов в удобном для пользователя виде.

Изучение дисциплины включает в себя: теоретический раздел (изучение теоретического материала); практический раздел (выполнение практических заданий); итоговый контроль результата изучения дисциплины. Данное пособие содержит в себе методические указания и варианты заданий для практических занятий, вопросы по организации самостоятельной работы.

2 Методические указания к проведению практических занятий

2.1 Практическое занятие «Исследование предметной области»

Цель работы

Целью данной работы является получение практических навыков исследования предметной области и сравнительного анализа существующих систем.

Рекомендации по подготовке к работе

Необходимо выбрать предметную область, с которой Вы будете работать в течение всех практических занятий в рамках практикума непроцедурного программирования.

Для успешной работы необходимо исследовать предметную область: определить существующие проблемы; рассмотреть автоматизированные системы, уже разработанные в данной области.

Для выбранной предметной области необходимо определить круг задач, которые должна решать система, которая будет разрабатываться Вами в процессе выполнения практических занятий.

Примерные темы предметных областей и возможный список решаемых задач приведены в таблице вариантов. Список решаемых задач может быть скорректирован: какие-то задачи могут быть добавлены или изменены Вами, по согласованию с преподавателем. Предметная область также может быть предложена студентом. При выборе предметной области студент должен прежде всего определить, на какие вопросы должна отвечать система, разрабатываемая в рамках этой области, для каких целей создается система.

Задание на практику

- 1) Выбрать предметную область. Выбор можно осуществить из предложенных ниже вариантов, или предложить свою и согласовать с преподавателем.
- 2) Исследовать выбранную предметную область на предмет существующих задач, проблем. Описать предметную область.
- 3) Сформулировать предполагаемый список вопросов, на которые должна отвечать система.

Варианты предметных областей

№ п/п	Предметная область	Возможный список решаемых задач
1	Расписание движения автобусов по Томской области (Томску).	<p>Определение оптимального маршрута из одной точки в другую (минимизация расстояния, времени или стоимости)</p> <p>- с указанием номеров автобусов и точками возможных пересадок</p>
2	Расписание авиаперелетов (по миру).	<p>Сформировать маршруты из X в Y, сортировать по стоимости, длительности. Учитывать допустимое время между пересадками.</p>
3	Расписание движения железнодорожного транспорта.	<p>Сформировать список маршрутов от начальной до конечной станции: время отправления, время прибытия</p> <p>Просмотр времени прохождения по маршруту</p>
4	Расписание приема врачей в поликлинике.	<p>Составить расписание врачей в поликлинике с указанием занятого (по талонам) и свободного времени: просмотр по запросу.</p> <p>Организовать возможность записи на прием к указанному врачу с корректировкой его занятости.</p> <p>Сбор статистики по врачам</p>
5	Работа риэлтерской компании.	<p>Поиск в системе требуемого варианта квартиры/дома/участка – по площади, по стоимости, по количеству квартир и т.п.</p> <p>Каждый формируемый список сортировать по возрастанию стоимости</p> <p>Формирование цепочки покупки/продажи недвижимости.</p>

№ п/п	Предметная область	Возможный список решаемых задач
6	Кулинария	Хранение, корректировка информации по производимым продуктам с указанием состава. Расчет стоимости любого продукта исходя из состава и стоимости составляющих. Возможность производства и расчета стоимости праздничных пирогов, тортов.
7	Учебный план направления.	Хранение информации. Корректировка информации с учетом ограничений по часам, количеству экзаменов и зачетов в семестре. Расчет нагрузок по семестрам. Формирование списков предметов по семестрам, по циклам.
8	Работа ресторана/кафе – меню, цены и т.п.	Просмотр меню по категориям (диетический, вегетарианский и т.п.), включая состав блюд Составление/выбор меню (на банкеты). Формирование списка блюд по заданным компонентам
9	Лекарственные растения.	Формировать список лекарственных растений, исходя из указанного заболевания. Сортировать список по доступности (ареал произрастания) либо по цене. Просмотр информации о конкретном растении
10	Библиотека (для изданий обязательно – год издания).	Выбор книги. Если книга на руках – время ожидания (когда будет сдана). Формирование-рекомендации по выбору книг с учетом рейтинга востребованности. Рейтинг – хранить и корректировать

№ п/п	Предметная область	Возможный список решаемых задач
11	Книжный магазин (для книг обязательно – цена).	Сформировать список книг по заданной области /автору. Списки сортировать по стоимости, году издания. Поиск книги по автору / году издания / части названия . Рекомендации подарочных книг.
12	Флористика.	Возможность составления клумб, бордюров, фасадов домов (с учетом цвета, высоты, долголетия растений и т.п.). Желательны рекомендации по дизайну.
13	Продажа автомобилей.	Рекомендации различных марок автомобилей исходя из потребностей и возможностей покупателя. Формирование и сортировка списков по заданным свойствам.
14	Аптека (лекарства – состав, стоимость, количество).	Формирование списка лекарств заданного назначения (учет наличия, стоимости, эффективности действия). Подсчет суммы сформированного заказа. Подсчет средней стоимости лекарств в заданной категории
15	Страхование (виды страхования, стоимость, длительность и т.п.).	Рекомендации и выбор требуемого вида страхования. Расчет стоимости страхования в зависимости от задаваемых условий. Определение возможных скидок. В случае досрочного расторжения определение возвращаемой суммы

№ п/п	Предметная область	Возможный список решаемых задач
16	Налогообложение физических лиц (зарплата, недвижимость ...).	Формирование списка всех видов налогообложения указанного лица. Учет различных льгот, освобождений от налогов, налоговых вычетов. В случае продажи недвижимости в течение года, подсчет суммы налога
17	Работа ЖКХ (виды услуг, стоимость).	Составление квитанций на оплату коммунальных услуг (по приборам учета, квадратуре квартиры). Список услуг ЖКХ, расписание, вызов мастеров.
18	Работа мебельного салона (продажа, изготовление).	Формирование прайс-листа на изготовление заданной мебели. Сформировать список моделей мебели по указанным параметрам. Сортировать список по цене. Изготовление – подсчет стоимости мебели по выбранным составляющим. Выбор готовых изделий заданной ценовой категории
19	Работа турагентства.	Сформировать список всех туров в указанную страну. Предложения по видам отдыха (пляжный, экскурсионный, активный ...) Поиск туров по заданной категории + желаемые даты.
20	Баскетбол (команды, соревнования, рейтинги).	Хранение, добавление, корректировка данных. Турнирные и рейтинговые таблицы.
21	Работа ателье по пошиву одежды.	Прайс-лист. Формирование счета на пошив одежды (с учетом работы, материала, фурнитуры). Сроки выполнения работ.

№ п/п	Предметная область	Возможный список решаемых задач
22	Шахматы	Организация игры – хранение текущей партии, выполнение ходов (с проверкой на корректность хода, с указанием шаха, мата).
23	Многофункциональный центр.	Виды услуг, часы работы. Запись на прием. Время выполнения работ – запись на получение готовых документов.
24	Биатлон (соревнования, рейтинги личные и командные).	Хранение, добавление, корректировка данных. Турнирные и рейтинговые таблицы. Корректировать: 1) рейтинг спортсменов по результатам новых соревнований; 2) рейтинг стран по рейтингам спортсменов и командным соревнованиям.
25	Кинофильмы (жанры, актеры, рейтинги...).	Выбор фильма (формирование списка) по жанру, рейтингу, актерам, режиссеру. Просмотр информации по актерам, режиссерам. Сортировка найденных фильмов по рейтингу/ году выпуска.
26	Банки (кредиты, вклады – расчеты).	По запросу на сумму и срок формировать список возможных кредитов с указанием ежемесячной и суммарной выплаты. Сортировать список по сумме. Определить сумму вкладов/ кредитов в банке. Рекомендации по вкладам/ кредитам.
28	Материально-техническая база предприятия (транспорт, здания/ сооружения, мебель, оборудование,...)	Хранение, добавление, корректировка данных. Формирование списка инвентаризации для заданной категории (с подсчетом общей суммы)

№ п/п	Предметная область	Возможный список решаемых задач
27	Кадровый состав предприятия (подразделения, штаты, оклады...).	<p>Формировать список свободных штатных единиц с сортировкой по величине оклада.</p> <p>Просмотр информации по каждому подразделению и предприятию в целом.</p> <p>Подсчет среднего оклада по предприятию/подразделению.</p> <p>Принятие на работу/ увольнение.</p>
29	Кинотеатр	<p>Определить список времени сеансов для заданного фильма (пон - , вторн -, ...). Подсчет прибыли на заданном отрезке времени.</p> <p>Подбор сеанса (жанр, стоимость, свободные места)</p> <p>Просмотр киносеансов за неделю /день. Поиск информации по заданному фильму.</p> <p>Подсчет рейтинга фильма по количеству проданных билетов.</p>
30	Музыка	<p>Поиск композиций по исполнителю/ группе/ автору. Сортировка списка по длительности/ году выпуска.</p> <p>Формирование альбома для прослушивания (время, жанры, исполнители).</p> <p>Возможность случайного выбора композиций.</p>
31	Продажа автозапчастей	<p>Формирование списка запчастей исходя из марки автомобиля, поломки и т.п.</p> <p>Сортировка списка запчастей по цене/ качеству.</p> <p>Формирование заказа с подсчетом суммарной стоимости.</p>
32	Футбол (соревнования, рейтинги личные и командные).	<p>Хранение, добавление, корректировка данных.</p> <p>Турнирные и рейтинговые таблицы.</p> <p>Сортировка данных</p>

№ п/п	Предметная область	Возможный список решаемых задач
33	Оптика	Продажа очков, оправ. Возможность заказа, формирование заказа, предложения, стоимость, сортировка предлагаемого списка вариантов

2.2 Практическое занятие «Определение системы»

Цель работы

Целью данной работы является знакомство с существующими моделями знаний, а также получение практических навыков сравнительного анализа существующих подходов к моделированию знаний.

Рекомендации по подготовке к работе

На предыдущем занятии Вами была выбрана предметная область и сформулированы задачи, которые должна решать создаваемая система.

На данном практическом занятии Вам необходимо разработать структуру данных/фактов/знаний. Для этого Вы должны изучить существующие модели представления знаний: фреймы, семантические сети, продукции и т.п. Например, для описания иерархических структур удобно использовать фреймы; для формального представления предметной области с множественными связями могут быть использованы семантические сети; для описания причинно-следственных связей подходят продукционные правила.

Уже на этапе выбора модели данных/знаний необходимо определиться с языком программирования, на котором Вы будете реализовывать систему: возможности и удобство языка могут повлиять на структуру представления данных.

Внимание! Если Вы выбираете не строго функциональный язык (многие языки являются гибридными и допускают и объектно-ориентированный подход), то помните, что **ОБЯЗАТЕЛЬНО** использовать технологии и возможности именно функционального программирования.

Задание на практику

- 1) Формально описать предметную область. В качестве формализма для описания могут быть выбраны таблицы, фреймы, семантические сети, деревья и др.
- 2) Выбрать язык программирования для создания системы.
- 3) Представить предметную область на одном из функциональных или логических языков. Создать базу данных в виде набора фактов и правил. Набор фактов должен быть достаточным, чтобы проверить работу вашей программы!

Требования к базе данных:

- 1) произведение количества полей на количество записей должно быть не менее 300 для табличного представления;
- 2) количество вершин в сети или листьев на дереве должно быть не менее 100 для сетевого и иерархического представления;
- 3) количество правил должно быть не менее 50 для продукционного представления.

2.3 Практическое занятие «Проектирование системы»

Цель работы

Целью данной работы является знакомство с использованием формальных методов конструирования программного обеспечения, а также получение практических навыков проектирования систем.

Рекомендации по подготовке к работе

На данном практическом занятии необходимо спроектировать разрабатываемую систему.

Продумайте алгоритмы, которые позволят решать поставленные задачи. При необходимости вы можете воспользоваться существующими алгоритмами просмотра, изменения, корректировки и сортировки данных.

Разработать и описать собственные алгоритмы, если это необходимо (проведения сложных расчетов, диалоговую модель работы системы и т.п.). Для описания алгоритмов можно использовать любые известные вам нотации.

Разработать сценарий диалога: какие вопросы сможет задавать пользователь; в каком виде будет вводиться/выводиться информация; в каком порядке будет задавать вопросы система и т.п.

Совет: по возможности, не заставляйте пользователя вводить длинные строковые данные: если количество вариантов вводимых данных ограничено, лучше выдавать список возможных вариантов и предлагать пользователю вводить только номер нужного варианта.

Задание на практику

- 1) Разработать и описать алгоритмы функционирования системы.
- 2) Разработать сценарий диалога системы с пользователем.

2.4 Практическое занятие «Разработка системы»

Цель работы

Целью данной работы является получение практических навыков разработки интеллектуальных систем с использованием возможностей функционального и логического программирования.

Рекомендации по подготовке к работе

В процессе данного практического занятия необходимо написать программу работы с БД фактов/данных/знаний, позволяющую решать весь список задач, который был Вами определен на первом занятии.

При реализации используйте алгоритмы, разработанные, или изученные Вами на предыдущем занятии. В процессе программирования можно использовать любые встроенные функции языка, любые конструкции и технологии. Приветствуется использование функций высокого порядка.

Диалог с системой может быть реализован в виде обычного консольного меню, либо с использованием графического интерфейса. Для реализации графического интерфейса можно использовать любой процедурный язык программирования (но только для интерфейса!).

Задание на практику

Создать программную реализацию системы, которая была спроектирована на первых трех занятиях.

2.5 Практическое занятие «Тестирование системы»

Цель работы

Целью данной работы является получение практических навыков тестирования разрабатываемых приложений.

Рекомендации по подготовке к работе

После разработки системы необходимо проверить корректность и полноту ее работы.

Проверьте, как будет вести себя системы, в случае неверных или ошибочных данных: система не должна аварийно заканчивать работу, а должна выдавать предупреждение о неверном вводе данных. В ситуации, когда вводятся данные, отсутствующие в БД (например, фамилия человека, о котором нет информации), система также должна предупреждать об этом. Более того, всегда должна быть возможность просмотреть данные, которые хранятся в системе.

Следующий шаг проверки – проверка полноты функционала с точки зрения поставленных задач. Необходимо протестировать весь список задач, определенных Вами еще на первом практическом занятии. Проверку осуществлять на множестве тестовых примерах, которые позволят в полной мере оценить корректность работы системы.

Задание на практику

Протестируйте работу созданной системы. Для этого

- 1) проверьте корректность работы системы в случае ввода неверных или ошибочных данных;
- 2) протестируйте правильность ответов системы на поставленные вопросы;
- 3) в случае обнаружения ошибок, исправить их.

3 Методические указания для организации самостоятельной работы

3.1 Общие положения

Цель самостоятельной работы по дисциплине – самостоятельное изучение теоретических разделов курса, подготовка к практическим занятиям, тестам.

Самостоятельная работа студента по дисциплине «Практикум непроцедурного программирования» включает следующие виды его активности:

1. изучение тем теоретической части дисциплины, вынесенных для самостоятельной проработки;
2. подготовка к практическим занятиям;
3. подготовка к тестам;
4. подготовка к зачету.

3.2 Самостоятельное изучение тем теоретической части курса

3.2.1 Модели представления знаний: фреймы

Необходимо рассмотреть существующие модели представления знаний: фреймы, семантические сети, продукции.

Для представления иерархических структур данных чаще всего используются фреймы. Существуют следующие модели фреймов: фреймы-структуры, фреймы-роли, фреймы сценарии, фреймы ситуации.

Фрейм имеет определенную внутреннюю структуру, состоящую из множества элементов, называемых слотами, которым также присваиваются имена. Каждый слот в свою очередь представляется определенной структурой данных. В значение слота подставляется конкретная информация, относящаяся к объекту, описываемому этим фреймом.

В состав слота входят такие компоненты, как имя, указатель наследования, указатель атрибутов, значение слота, демон.

Важнейшим свойством теории фреймов является наследование свойств. Во фреймах наследование происходит по АКО-связям (A-Kind-Of = это). Слот АКО указывает на фрейм более высокого уровня иерархии, откуда неявно наследуются, то есть переносятся, значения аналогичных слотов.

3.2.2 Модели представления знаний: семантические сети

Одной из часто применяемых моделей является конструкция, названная семантической сетью. Семантический подход к построению систем искусственного интеллекта находит применение в системах понимания естественного языка, в вопросно-ответных системах, в различных предметно-ориентированных системах.

Семантические сети широко используются в экспертных системах в качестве языка представления знаний, в системах распознавания речи и понимания естественного языка.

В самом общем случае семантическая сеть представляет собой информационную модель предметной области и имеет вид графа, вершины которого соответствуют объектам предметной области, а дуги – отношениям между ними.

3.2.3 Обработка и хранение знаний: свойства символов, ассоциативные списки.

Для обработки и хранения знаний в Лиспе существует два типа списков: списки свойств и ассоциативные списки.

В Лиспе с символом можно связывать, не только значение, но и информацию, называемую списком свойств (property list).

Списки свойств широко используются для представления смысловой информации, когда с атомом необходимо связать список определенных свойств и связанных с ними значений. Существуют специальные функции языка, позволяющие выполнять различные действия с такими списками: присвоить, прочитать, заменить, удалить свойство. Кроме этого, всегда можно просмотреть информацию о списке свойств.

Ассоциативный список или просто а-список (a-list) есть основанная на списках и точечных парах структура данных, описывающая связи наборов данных и ключевых полей, для работы с которой существуют готовые функции.

Ассоциативные списки применяются при работе с динамическими базами данных в оперативной памяти. Ассоциативный список можно рассматривать как отображение множества ключей на множество соответствующих им объектов. Для работы с такими списками существуют функции создания ассоциативного списка и поиска элементов в ассоциативном списке. К сожалению, не существует готовой функции, позволяющей изменить значение ключа в таком списке. При необходимости функцию изменения значений в ассоциативном списке необходимо писать самостоятельно.

3.2.4 Использование технологий функционального программирования для создания интеллектуальных систем

В связи со стремительным ростом машинного обучения (МО) и больших данных функциональное программирование стало набирать популярность из-за простоты распараллеливания чистых функций. Функциональная парадигма также упрощает отслеживание, тестирование и обслуживание кода для задач анализа данных и рабочих процессов, что создает предпосылки для ее активного использования в ближайшем будущем.

Благодаря чистоте своих функций функциональное программирование отлично подходит для решения таких актуальных задач, как МО и анализ данных. Однако это совсем не значит, что вам следует прекратить писать на других языках программирования и обходиться только функциональное программирование, поскольку у каждого из них есть спектр задач, с которыми они справляются лучше других. При этом крайне важно усвоить основные принципы, чтобы по мере необходимости вы могли эффективно их применять. Именно по этим причинам функциональное программирование ассоциируется с будущим, хотя, возможно, по большей части с будущим МО и больших данных, чем разработки ПО в целом.

Особенности функционального программирования: применение чистых функций; оптимальная прозрачность; статические переменные; удобное параллельное программирование; проверка сигнатур функций.

3.2.5 Тестирование программного обеспечения

Функциональное испытание — это тип тестирования программного обеспечения, который проверяет программную систему на соответствие функциональным требованиям / спецификациям. Цель функциональных тестов состоит в том, чтобы проверить каждую функцию программного приложения, предоставляя соответствующий ввод, проверяя выход в соответствии с функциональными требованиями.

Основной целью функционального тестирования является проверка функциональности системы программного обеспечения. Он в основном концентрируется на —

- Основные функции: тестирование основных функций приложения
- Базовое удобство использования: включает базовое тестирование системы. Он проверяет, может ли пользователь свободно перемещаться по экранам без каких-либо затруднений.
- Доступность: проверяет доступность системы для пользователя.

- Условия ошибки: Использование методов тестирования для проверки ошибок. Он проверяет, отображаются ли подходящие сообщения об ошибках.

4 Рекомендуемая литература

1. Салмина Н. Ю., Функциональное программирование и интеллектуальные системы: учебное пособие [Электронный ресурс] / Салмина Н. Ю. — Томск: ТУСУР, 2016 . — 100 с. — Режим доступа: <https://edu.tusur.ru/publications/63572>).
2. Цуканова, Н.И. Теория и практика логического программирования на языке Visual Prolog 7. Учебное пособие для вузов [Электронный ресурс] : учеб. пособие / Н.И. Цуканова, Т.А. Дмитриева. — Электрон. дан. — Москва: Горячая линия-Телеком, 2013. — 232 с. — ЭБС ЛАНЬ. — Режим доступа: <https://e.lanbook.com/book/11847>.