

МИНОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ

Директор института прикладной
математики и компьютерных наук

А.В. Замятин

« 14 » *ноября* 2021 г.



Фонд оценочных средств по дисциплине

Структурное проектирование

Направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

код и наименование направления подготовки

Искусственный интеллект и разработка программных продуктов

наименование профиля подготовки

ФОС составил(и):

канд. техн. наук, доцент,
доцент кафедры программной инженерии

А.М. Бабанов

Рецензент:

д-р физ.-мат. наук, профессор,
профессор кафедры программной инженерии

О.А. Змеев

Фонд оценочных средств одобрен на заседании учебно-методической комиссии
института прикладной математики и компьютерных наук (УМК ИПМКН)

Протокол от 17 июня 2021 г. № 05

Председатель УМК ИПМКН,
д-р техн. наук, профессор

С.П. Сущенко

Фонд оценочных средств (ФОС) является элементом системы оценивания сформированности компетенций у обучающихся в целом или на определенном этапе ее формирования.

ФОС разрабатывается в соответствии с рабочей программой (РП) дисциплины и включает в себя набор оценочных материалов для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине.

1. Компетенции и результаты обучения, формируемые в результате освоения дисциплины

Компетенция	Индикатор компетенции	Код и наименование результатов обучения (планируемые результаты обучения, характеризующие этапы формирования компетенций)	Критерии оценивания результатов обучения			
			Отлично	Хорошо	Удовлетворительно	Неудовлетворительно
ОПК-4. Способен участвовать в разработке технической документации программных продуктов и программных комплексов	ИОПК-4.1. Обладает необходимыми знаниями нормативной базы профессиональной деятельности	ОР-4.1.1. Знать методы структурного проектирования	Имеет общее представление о методах структурного проектирования, знает особенности их применения	Имеет общее представление о методах структурного проектирования	Имеет слабое представление о методах структурного проектирования	Не имеет представления о методах структурного проектирования
		ОР-4.1.2. Знать метод SADT	Имеет общее представление о методе SADT, знает особенности его применения	Имеет общее представление о методе SADT	Имеет слабое представление о методе SADT	Не имеет представления о методе SADT

	ИОПК-4.2. Применяет знания нормативной базы в профессиональной деятельности	ОП-4.2.1. Уметь проектировать системы БД методами структурного проектирования	Умеет безошибочно проектировать системы БД методами структурного проектирования	Умеет проектировать системы БД методами структурного проектирования, но допускает незначительные ошибки	Допускает серьезные ошибки при проектировании систем БД методами структурного проектирования	Не умеет проектировать системы БД методами структурного проектирования
	ИОПК-4.3. Разрабатывает техническую документацию, связанную с профессиональной деятельностью	ОП-4.3.1. Уметь строить SADT-модели	Умеет безошибочно строить SADT-модели	Умеет строить SADT-модели, но допускает незначительные ошибки	Допускает серьезные ошибки при построении SADT-моделей	Не умеет строить SADT-модели

2. Этапы формирования компетенций и виды оценочных средств

Компетенции, формируемые в результате обучения дисциплине «Дискретная математика», при текущем контроле проверяются все сразу на основе материала изучаемых в течение семестра тем лекционных и практических занятий.

№	Этапы формирования компетенций (разделы дисциплины)	Код и наименование результатов обучения	Вид оценочного средства (тесты, задания, кейсы, вопросы и др.)
1	Контролируется на основе материала Темы 2, Темы 3, Темы 4, Темы 5	ОПК-4.	1. Лабораторная работа №1. 2. Лабораторная работа №2.
2	Контролируется на основе материала Те-мы 4, Темы 5, Темы 6, Темы 7	ОПК-4.	1. Лабораторная работа №2. 2. Лабораторная работа №3. 3. Лабораторная работа №4.
3	Контролируется на основе материала Те-мы 4, Темы 5, Темы 6, Темы 7	ОПК-4.	1. Лабораторная работа №2. 2. Лабораторная работа №3. 3. Лабораторная работа №4.

3. Типовые контрольные задания или иные материалы, необходимые для оценки образовательных результатов обучения

3.1. Типовые задания для проведения текущего контроля успеваемости по дисциплине

Требования к лабораторным работам

Требования к лабораторной работе 1:

1. Построить SADT-модели всех организационных (выполняемых людьми, организациями, подразделениями) бизнес-процессов, выделенных в техническом задании. Если существует бизнес-процесс, частями которого являются бизнес-процессы технического задания, можно представить одну общую SADT-модель.

Модели должны отражать новое видение процессов с участием будущей информационной системы (ИС).

Листовые блоки модели с участием ИС должны соответствовать одной логической транзакции (единой непрерываемой единице диалога, выполняющей законченную работу с данными одного человека).

Для стыковки организационных SADT-моделей с будущими моделями функций ИС (DFD, FHD) можно либо использовать одинаковые наименования блоков SADT и функций, либо указать наименования функций ИС в качестве специфических механизмов SADT-блоков.

Можно (но это не является обязательным) придумать модели, отражающие старое видение процессов (как бы до реинжиниринга).

2. Утвердить SADT-модели бизнес-процессов (на бумаге) у преподавателя (в дальнейшем необходимо иметь их при себе при любых контактах с преподавателем по поводу выполнения последующих лабораторных работ).

Требования к лабораторной работе 2

1. Подготовить и привести в соответствие информационные схемы предметной области (как в репозитории, так и в словаре Oracle), построенные в ходе лабораторных работ по курсу «Базы данных».

В частности должны быть обеспечены:

- ER-схема в нотации Oracle Designer (Баркера);
- реляционная схема в репозитории и словаре Oracle, включающая определения таблиц, представлений, последовательностей, триггеров и ограничений целостности.

Чтобы было меньше проблем в дальнейшем, в идеале следует повторить весь процесс проектирования БД в прямом направлении (от ERD до реляционных схем), максимально добиваясь автоматизма преобразований (DDT, генератор DE) и минимизируя ручную доводку.

При этом в обязательном порядке (опять же чтобы не было проблем в дальнейшем) в каждой таблице обязан быть первичный ключ. Он должен быть либо суррогатным ключом, либо (в случае таблиц, реализующих связи M:N) – группой из двух внешних ключей, ссылающихся на суррогатные ключи.

2. Построить РМ-модель предметной области, определив функциональные роли всех пользователей и процессы (функции), соответствующие логическим транзакциям.

Декомпозицию процессов в РМ следует проводить до тех пор, пока не будут получены функции, целиком выполняемые пользователем одной роли. Если при этом они не будут элементарными (реализуемыми с помощью одного окна диалога), дальнейшую декомпозицию лучше провести в DFD или FHD.

3. Утвердить РМ-модель (на бумаге) у преподавателя (в дальнейшем необходимо иметь ее при себе при любых контактах с преподавателем по поводу выполнения последующих лабораторных работ).
4. Для сложных функций (включающих не одно окно диалога) построить DFD-модели.
5. Для всех функций РМ-модели построить FHD-модели.

Листьями этих моделей должны быть функции, чьи задачи решаются с помощью одной формы (возможно с подформами).

Если существует бизнес-процесс, частями которого являются бизнес-процессы технического задания, можно представить одну общую FHD-модель.

Для функций-листьев указать использование множеств сущностей и атрибутов ERD-модели.

Полезно перед этим представить эскиз окна, определить master и detail компоненты модуля, для каждого компонента модуля определить base и, возможно, lookup таблицы. (Для знакомства с особенностями диалога и структурой OF-модулей обратитесь к файлу "OD OF Generator.doc" в каталоге Лаборатории\DOCS.)

Использование данных в функциях определяется пока на уровне множеств сущностей ER-модели и их атрибутов. Поскольку сейчас ничего нельзя сказать об использовании множеств связей, действия с ними необходимо не забыть указать позже на уровне использования таблиц в модулях.

6. Утвердить DFD- и FHD-модели (на бумаге) у преподавателя (в дальнейшем необходимо иметь ее при себе при любых контактах с преподавателем по поводу выполнения последующих лабораторных работ).

7. Построить в Matrix Diagrammer (MD) и применять для контроля использования данных матрицы “Business Functions to Entities” и “Business Functions to Attributes”, а для контроля использования функций - “Business Units to Business Functions”.

Требования к лабораторной работе 3

1. Для освоения инструментов Oracle Designer и методики проектирования и генерации интернет-приложений в виде интерактивных форм с применением Oracle Forms Developer предлагаются следующие источники информации:
 - Колетски П., Дорси П. Oracle Designer. Настольная книга пользователя. – М.:, Изд-во "ЛОРИ", 1999. – 592 с. (Избранные главы этой книги есть в электронной библиотеке факультета информатики).
 - Online Help Oracle Designer.
 - Бабанов А.М. Технология разработки программного обеспечения: структурный подход: Учебное пособие. – Томск: Изд-во НТЛ, 2006. – 220 с. (Достаточно большое количество экземпляров этой книги есть в Научной библиотеке ТГУ как в читальном зале, так и на абонементе).
 - Файл OD OF Generator.doc в каталоге Лаборатории\DOCS.
2. Используя Application Design Transformer создать набор модулей-кандидатов из данных анализа (Function Hierarchy Diagrammer или Dataflow Diagrammer). Параметры генерации:
 - Generate Modules = Modules;
 - выберите Start Function, которая будет использоваться как корень иерархии функций; чтобы сгенерировать все функции в пределах текущей прикладной системы, оставьте это поле пустым; для определения Start Number (номера генерируемого модуля) при генерации отдельного модуля используйте кнопку Find highest.
 - выберите язык модуля Screen = Oracle Forms, Report = Oracle Reports, Utility = PL/SQL
3. Используя RON (Repository Object Navigator) или DE (Design Editor) примите или отклоните модули-кандидаты. Для этого установите свойство модулей Candidate? в ‘No’ или удалите их. Любые модули-кандидаты, которые Вы не примете, останутся модулями-кандидатами.
4. Внесите коррективы в значения свойств модулей – кандидатов. В названиях модуля (свойства Name и Implementation Name) должны использоваться только латинские символы и их длина не должна быть более 8 знаков.
5. Спланируйте структуру модулей. С помощью Design Editor создайте определение модулей (Module Diagrams).
6. Генерация Oracle Forms-приложения:
 - открыть DE;
 - в окне Navigator выбрать закладку Modules;
 - правой кнопкой мыши щелкнуть по генерируемому модулю;
 - выбрать альтернативу меню Generate...;
 - в разделе Preserve Layout должен быть выбран вариант «No», если вы создаете новое приложение и вариант «Yes» с указанием в поле Source Form Name файла уже созданного приложения;

- настроить опции генерации (кнопка Options...)
на вкладке Dependencies в разделе Source используются настройки по умолчанию;
 - на вкладке Form Options в поле Destination of generated files указать путь до папки, куда генерируется ваше приложение;
 - на вкладке Compile параметр Compile Form/Menu = No;
 - на вкладке Run параметр Run form automatically=No;
 - остальные параметры по умолчанию.
 - выполнить генерацию – кнопка Start.
7. Компиляция и просмотр Oracle Forms-проекта выполняется в среде Oracle Forms Developer:
 - Перед использованием Oracle Forms запустить Forms Developer → Start OC4J Instance;
 - Компиляция модулей осуществляется в Forms Builder (Forms Developer →Forms Builder);
 8. Компиляция модулей в Forms Builder:
 - При первичном использовании Forms Builder необходимо установить браузер, который будет использоваться для просмотра и работы с формами приложения Edit → Preferences... → Runtime → Web Browser Location = C:\Program Files\Mozilla Firefox\firefox.exe;
 - Открыть нужный модуль File → Open;
 - Соединиться с БД File → Connect.. → указать параметры соединения;
 - Выполнить компиляцию модуля Program → Compile Module;
 - Запустить приложение Program → Run Form.
 9. Для всех модулей построить адекватные диаграммы данных и диалога.
 10. Воспользовавшись предлагаемой методикой, разработать предварительный проект модулей и отдельные черновые Oracle_Forms-приложения для них. Они должны обеспечивать полное функционирование ИС, возможно без особого блеска и интеллекта.

Требования к лабораторной работе 4

1. Систематическое изложение методики совершенствования OF-приложений дается в
 - Файл OD OF Generator.doc в каталоге Лаборатории\DOCS.
2. Воспользовавшись предлагаемой методикой, разработать окончательный проект модулей и законченное единое Oracle_Forms - приложение для индивидуальной предметной области.
Оно должно обеспечивать полное функционирование ИС с идеальным диалогом и интеллектом (ничего лишнего, все по делу и удобно).
Можно сгенерировать несколько приложений (Oracle_Forms - проектов) для каждой роли пользователей по отдельности.
3. Для правильной организации связи между модулями вашего приложения, все сгенерированные модули должны находиться в папке C:\DATA\oracle\forms. В рамках одного сеанса работы модули генерируются в данную папку. По окончании сеанса все результаты работы копируются на личный диск.
4. В обязательном порядке необходимо использовать:

- зоны с разными стилями расположения;
- широкий спектр типов элементов управления;
- группы элементов управления;
- сложенные (stacking) области с закладками;
- диалоги свойств;
- диалоги запросов;
- диалоги LOV;
- экран запуска;
- значения по умолчанию;
- горячие клавиши;
- кнопки;
- подсказки (hints);
- меню.

4. Методические материалы, определяющие процедуры оценивания образовательных результатов обучения

4.1. Методические материалы для оценки текущего контроля успеваемости по дисциплине.

Для оценки лабораторных работ используется расширенная шкала оценивания, приведенная ниже в таблице.

Оценка	Форма записи прописью	Численное значение	Критерий оценивания	Перевод в традиционную шкалу
5+	Отл-плюс	5,3	Обучающийся показал творческое отношение к обучению, в совершенстве овладел всеми теоретическими вопросами, показал все требуемые умения и навыки в работе с программными продуктами.	Отлично
5	Отлично	5,0	Обучающийся показал отличный уровень владения всеми теоретическими вопросами, показал все требуемые умения и навыки в работе с программными продуктами.	
5-	Отл-минус	4,7		
4+	Хор-плюс	4,3	Обучающийся овладел всеми теоретическими вопросами, частично показал основные умения и навыки в работе с программными продуктами.	Хорошо
4	Хорошо	4,0		
4-	Хор-минус	3,7		
3+	Уд-плюс	3,3	Обучающийся овладел всеми теоретическими вопросами, частично показал основные умения и навыки в работе с программными продуктами.	Удовлетворительно
3	Удовл.	3,0		

3-	Уд-минус	2,7	Обучающийся имеет недостаточно глубокие знания по теоретическим разделам дисциплины, показал не все основные умения и навыки в работе с программными продуктами. Минимально возможный допустимый уровень владения предметом.	Неудовлетворительно
2+	Неуд-плюс	0	Обучающийся имеет существенные пробелы по отдельным теоретическим разделам дисциплины и не владеет основными умениями и навыками в работе с программными продуктами, но с возможностью повторной пересдачи экзамена	
2	Неудовл.	0	Обучающийся имеет существенные пробелы по отдельным теоретическим разделам дисциплины и не владеет основными умениями и навыками в работе с программными продуктами, требуется повторное изучение дисциплины	

4.2. Методические материалы для проведения промежуточной аттестации по дисциплине.

Оценка за промежуточную аттестацию по дисциплине выставляется как среднеарифметическая по итогам текущего контроля успеваемости при условии сдачи контрольной точки 4 (лабораторная работа 4) на положительную оценку.

Для оценки промежуточной аттестации используется традиционная шкала оценивания. Перевод из расширенной шкалы в традиционную приведен в таблице выше.