

РАБОЧАЯ ПРОГРАММА ВВОДНОЙ УЧЕБНОЙ ДИСЦИПЛИНЫ «Основы промышленной разработки программного обеспечения»

Количество часов – академических 60 (40 часов лекций и 20 часов практики)

Число занятий – 20 по 3 академических часа

Темы занятий и краткое содержание

Лектор Ржавин В.В.

Занятие 1

Тема 1. Введение в управление разработкой ПО

Лекция 1. Проект. Цель проекта. Управление проектом. Иерархическая структура работ по проекту (ИСР). Принципы декомпозиции. Методика построения иерархической структуры.
Факторы успеха и провала проектов. Участники процесса разработки ПО. Сложности управления проектом. Понятие жизненного цикла (ЖЦ). Этапы ЖЦ. Стандартные модели ЖЦ. Каскадная схема разработки ПО. Спиральная модель ЖЦ ПО. Обзор современных методологий разработки ПО (XP, Agile, Scrum, MSF для СММІ). Современные подходы к разработке ПО.

Самостоятельная работа. Описать цели проекта. Например: автоматизировать процесс сбора показаний, чтобы снизить затраты времени на ввод и учет показаний.
Создать иерархическую структуру работ по проекту состоящую из этапов проекта (анализ требований, проектирование, реализация, тестирование).

Тема 2. Введение в разработку требований

Лекция 2. Что такое требования и зачем нужны требования. Требования к продукту и проекту. Бизнес требования. Функциональные и нефункциональные требования. Системные требования. Характеристика (feature). Взаимоотношения между функциями, пользовательским и функциональными требованиями.
Пользовательские истории (User Story) — способ описания требований к разрабатываемой системе.

Занятие 2

Лекция 3. Нефункциональные требования. Характеристики качества. Характеристики качества. Классификация К.Вигерса. Классификация требований по уровню. Нефункциональные требования (по К.Вигерсу). Бизнес правила. Атрибуты (характеристики) качества. Внешние интерфейсы. Ограничения. Свойства требований. Прототипирование.

Практика 1. Формирование требований. Слушатели знакомятся с единой для всех предметной областью – «Регистрация показаний домашних счетчиков».

Преподаватель выступает в роли заказчика проекта. Вместе со слушателями формируются начальные бизнес-требования, пользовательские требования, функциональные требования. Приводятся примеры нефункциональных требований. Обсуждаются системные требования. Вводится понятие трассировки требований.

Самостоятельная работа. Слушателям предлагается на основе предложенных шаблонов продолжить работу по формированию требований для заданной предметной области. Каждое требование должно удовлетворять заданным свойствам.

1. Разрабатываются бизнес-требования и пользовательские требования (Например: Пользователь должен иметь возможность сохранять показания прибора учета за различные месяцы, чтобы посчитать потребление за выбранный период времени).
2. Каждый слушатель должен создать графическую спецификацию прототипа пользовательского интерфейса для заданной предметной области в виде графического файла или рисунка в MS Word документе.
3. Описываются ограничения проекта (Например, Программа должна работать под OS Windows 7 и Windows 10, данные показаний должны храниться в СУБД SQL Server 2012)

Занятие 3

Тема 3. Объектно-ориентированный анализ и проектирование

Лекция 4. Диаграмма UC. Переход от ТЗ к диаграммам вариантов использования (ДВИ) и нефункциональным требованиям. Цели разработки ДВИ. Основные элементы диаграммы. Структурирование вариантов использования. Уровни прецедентов. Отношения на диаграммах использования. Ассоциация. Структуризация прецедента. Обобщение. Включение. Расширение. Порядок построения диаграммы. Правила составления прецедентов. Когда составляются прецеденты и насколько детально. Список основных правил.
Диаграмма деятельности. Отображения рабочего процесса между разными вариантами использования. Элементы и нотация диаграммы деятельности. Отображение прецедента на диаграмму деятельности.

Лекция 5. Реализация прецедентов. Моделирование при помощи диаграмм прецедентов. Документирование потока событий. Реализация диаграммами деятельности. Реализация диаграммами взаимодействия. Распространенные ошибки при моделировании прецедентов. Советы для написания хороших сценариев использования.

Самостоятельная работа. Слушатели должны:

1. Создать UC-диаграмму согласно разработанным пользовательским требованиям.
2. Описать бизнес-процесс с помощью диаграммы деятельности (UML Activity Diagram). (Например: процесс ввода показаний прибора учета в первый раз, включая добавление нового лицевого счета, нового прибора учета, ввода показаний для прибора учета для выбранного месяца)

Занятие 4

Лекция 6. Модель анализа. Объектная декомпозиция. Объектные аналитические модели: модель классов, модель взаимодействия, модель состояния.

Модель проектирования. Предварительное проектирование (архитектурный уровень) и детальное проектирование (алгоритмический уровень).

Архитектурное проектирование – структурирование системы, моделирование управления, декомпозиция подсистем на модули.

Практика 2. Для проекта «Регистрация показаний домашних счетчиков» совместными усилиями слушателей строится сценарий одного прецедента *UC*-диаграммы. Рассматриваются как основной, так и альтернативные сценарии. На этом примере разбираются основные ошибки документирования сценариев.

Самостоятельная работа. Слушатели должны:

1. Для каждого прецедента расписать сценарий его использования, используя предложенный шаблон.
2. Описать функциональные требования (например: Программа должна давать возможность сохранять информацию по владельцу лицевого счета).
3. Провести трассировку требований: функциональные vs бизнес-требования, бизнес-требования vs варианты использования, функциональные vs сценарии использования.
4. Описать нефункциональных требований (например: Программа должна выполнять расчет среднемесячного потребления за год за время не более 10 секунд).

Занятие 5

Лекция 7. Диаграмма компонентов. Компоненты и артефакты. Понятие компонента. Нотация диаграммы компонентов: компоненты, интерфейсы, зависимости между компонентами и интерфейсами. Стандартные стереотипы компонентов. Отображение портов компонента. Ссылки между компонентами. Отображение внутренних частей компонента.

Лекция 8. Диаграмма классов. Назначение диаграммы классов. Элементы ДК. Класс и объекты. Варианты графического изображения класса. Синтаксис. Типичная нотация класса. Атрибуты класса. Видимость, область действия, кратность атрибута. Операции: синтаксис, имя, список параметров, направление передачи параметров. Типы операций. Виды отношений между классами. Нотация UML для отображения взаимосвязи между классами. Виды классов. Конкретные и абстрактные классы. Интерфейс. Расширение языка UML.

Занятие 6

Лекция 9. Диаграммы последовательности. Диаграмма последовательности (sequence diagram). Нотация диаграммы последовательности (ДП). Объекты, сообщения, линия жизни, фокус управления на ДП. Ветвление потока управления. Правила построения.

Практика 3. Приводятся примеры построения диаграмм компонентов, диаграмм классов и диаграмм взаимодействия.

Самостоятельная работа.

1. Каждый слушатель должен в соответствии со своим видением проектируемой системы описать модули программы с помощью диаграммы компонентов. Необходимо описать назначение каждого компонента и его взаимодействия на диаграмме.
2. Описать диаграмму классов для одного и модулей программы.
3. Описать взаимодействие объектов с помощью диаграммы последовательностей (например, расписать сценарий сохранения показаний с формы приложения в СУБД).

Занятие 7

Тема 4. Введение в тестирование

Лекция 10. Тестирование. Цель тестирования. Тестирование, верификация и валидация. Модель взаимодействия разработчиков ПО и тестировщиков. Принципы тестирования. Документация, создаваемая на различных этапах ЖЦ. Место специалистов по тестированию в процессе разработки ПО.

Лекция 11. Тестирование методом черного ящика. Стратегии черного ящика: эквивалентное разбиение; анализ граничных значений; применение функциональных диаграмм; предположение об ошибке.

Занятие 8

Лекция 12. Что такое тест-кейс (ТС). Структура тест-кейса. Что такое баг-репорт. Стандартные ошибки при оформлении тест-кейсов.

Практика 4. Рассматриваются примеры тест-кейсов. Знакомство с шаблоном ТС. Начинается разработка набора тестов для проекта «Регистрация показаний домовых счетчиков».

Самостоятельная работа. Закончить работу по составлению набора ТС.

Занятие 9

Тема 5. Проектирование баз данных

Лекция 13. Основы проектирование реляционных баз данных. Знакомство с основными нотациями. Построение ER-диаграммы. Правила нормализации.

Практика 5. Построение схемы базы данных для проекта «Регистрация показаний домовых счетчиков».

Самостоятельная работа. Внести дополнительную функциональность в проект и расширить предложенную схему базы данных.

Занятие 10

Лекция 14. Диаграмма состояний (ДС). Назначение и применение. Графическое изображение состояний на ДС. Виды состояний. Виды событий: событие вызова, событие сигнала, событие таймера, событие изменения.

Типы приложений. Настольные приложения. Веб-приложения. Веб-приложения с богатым пользовательским интерфейсом (RIA). Веб-сервисы. Мобильные приложения.

Лекция 15. Оценка стоимости (сложности) программного продукта. Модель оценки стоимости COSOMO, методы экспертной оценки, покер планирования, PERT.

Самостоятельная работа.

1. Описать состояния с помощью диаграммы состояний. (Например, описать состояния формы запроса показаний за выбранный период: “запрос выполняется”, “запрос выполнен”, “ввод данных запроса”).
2. Декомпонировать изменения в каждом модуле на диаграмме компонентов на постановки задач с оценками трудозатрат.
3. Произвести расчет суммарных трудозатрат на реализацию каждого функционального требования.

Лектор Лобастова О.А.

Занятие 11

Лекция 1. Принцип работы приложений .NET. Среда разработки Visual Studio 2015. Создание разных видов приложений. Решения (Solution). Проект (Project). Компиляция. Отладка. Функция Main приложения. Методы и аргументы методов. Переменные и константы, глобальные и локальные данные. Область видимости переменных, стек вызовов метода. Значимые и ссылочные типы данных. Математические операции. Префиксные и постфиксные операции. Операции отношения. Приоритет операций. Обработка ошибок приложения (Exception, try/catch/finally).

Практика 1. Разработка консольного приложения, считывающего два целочисленных параметра из командой строки и выводящее их сумму, разность и произведение.

Самостоятельная работа. Изменение программы для работы как с целочисленными, так и с вещественными параметрами командной строки.

Занятие 12

Лекция 2. Установка и настройка Microsoft SQL Server. Аутентификация, учетные записи пользователей. Создание базы данных. Создание таблиц и представлений. Обеспечение целостности данных (ограничения NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK...). Резервное копирование и восстановление.

Практика 2. Физическая реализация базы данных «Показания счетчиков» на примере двух связанных таблиц.

Самостоятельная работа. Добавление всех необходимых таблиц в базу данных.

Занятие 13

Лекция 3. Одномерные и многомерные массивы. Инициализация массивов. Перечисления. Структуры. Списки (List) и словари (Dictionary). Операторы условного и безусловного перехода. Операторы цикла и перебора (foreach). Файловый ввод-вывод.

Практика 3. Разработка консольного приложения, считывающего из файла массив данных и выводящего максимальный, минимальный элементы и среднее значение. Разработка структуры для хранения показаний счетчиков.

Самостоятельная работа. Изменение программы для считывания данных в структуру.

Занятие 14

Лекция 4. Редактор запросов SQL Server Management Studio. Инструкция SELECT. Логическая обработка запросов. Предложения FROM, WHERE, GROUP BY, HAVING, ORDER BY. Агрегатные функции.

Практика 4. Создание запросов на выборку данных с использованием группировки и агрегатных функций.

Самостоятельная работа. Разработка собственных запросов к базе данных.

Занятие 15

Лекция 5. Объединения: LEFT JOIN, INNER JOIN, RIGHT JOIN. Подзапросы. Представления.

Практика 5. Создание сложных запросов к базе данных. Разработка представлений.

Самостоятельная работа. Добавление собственных представлений в базу.

Занятие 16

Лекция 6. Классы и их атрибуты (поля, свойства, события, конструктор, деструктор, методы, понятие инкапсуляции и области видимости атрибутов private и public). Принципы работы приложений Windows. Графический интерфейс пользователя. Формы. Основные элементы управления и компоненты. События.

Практика 6. Разработка GUI-приложения, считывающего данные из текстовых полей в элементы структуры. Разработка классов для хранения показаний счетчиков.

Самостоятельная работа. Доработка классов. Изменение программы для считывания данных в экземпляры классов.

Занятие 17

Лекция 7. Вставка данных. Обновление данных. Удаление данных.

Практика 7. Создание запросов на добавление, удаление и изменение данных.

Самостоятельная работа. Разработка собственных запросов на добавление, удаление и изменение данных.

Занятие 18

Лекция 8. Хранимые процедуры и функции. Курсоры. Триггеры.

Практика 8. Создание хранимой процедуры в базе данных «Показания счетчиков»

Самостоятельная работа. Разработка собственных хранимых процедур и/или функций.

Занятие 19

Лекция 9. Технология ADO.NET. Поставщики данных. Извлечение данных с помощью DataReader.

Практика 9. Изменение GUI-приложения для сохранения данных в базу данных на примере одной таблицы.

Самостоятельная работа. Доработка программы для сохранения данных всех таблиц базы данных.

Занятие 20

Лекция 10. Набор данных DataSet. Адаптер данных DataAdapter. Связывание, просмотр и фильтрация данных.

Практика 10. Доработка GUI-приложения для вывода данных из базы данных.

Самостоятельная работа. Добавление фильтров, сортировки, поиска. Тестирование программы.