

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

Факультет вычислительной техники
Кафедра «Информационная безопасность»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.В.06 «Организация и методы программирования»

Специальность: 10.05.03 Информационная безопасность автоматизированных
систем

Специализация: № 8 Разработка автоматизированных систем в защищенном
исполнении

ОПОП по специальности:

Информационная безопасность автоматизированных систем

Квалификация выпускника: специалист по защите информации

Форма обучения - очная

Срок обучения — 5,5 лет

Форма обучения - очная

Срок обучения — 5,5 лет

1 ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов (практических заданий, описаний форм и процедур проверки), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы (ОПОП).

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций обучающихся целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных и профессиональных компетенций и индикаторов их достижения, приобретаемых обучающимся в соответствии с требованиями ОПОП.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости и промежуточная аттестация проводятся с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся, организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся на практических занятиях по результатам выполнения и защиты обучающимися индивидуальных заданий, по результатам выполнения контрольных работ и тестов, по результатам проверки качества конспектов лекций и иных материалов.

В качестве оценочных средств на протяжении семестра используется устные и письменные ответы студентов на индивидуальные вопросы, письменное тестирование по теоретическим разделам курса. Дополнительным средством оценки знаний и умений студентов является отчет о выполнении практических заданий их защита.

По итогам курса обучающиеся сдают экзамен и выполняют курсовую работу. Форма проведения экзамена – устный ответ с письменным подкреплением по утвержденным билетам, сформулированным с учетом содержания дисциплины. В билет для зачета включается два теоретических вопроса и задача. В процессе подготовки к устному ответу студент должен составить в письменном виде план ответа.

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
1	2	3	4
1.	Введение	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
2.	Основные технологии разработки ПО	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
3.	Основы объектно-ориентированного программирования (ООП)	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
4.	Основы технологии проектирования программного обеспечения (ТППО)	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
5.	Основы языка визуального моделирования UML (Unified Modeling Language)	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
6.	Контроль программ. Отладка и тестирование ПО	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
7.	Структуры данных в программном обеспечении ЭВМ	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен
8.	Методы обработки данных в вычислительных системах	ПК-3 (ПК-3.1, ПК-3.2)	Экзамен

2 ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ

При освоении дисциплины формируются следующие компетенции: ПК-3 (ПК-3.1, ПК-3.2).

Указанные компетенции формируются в соответствии со следующими этапами:

- формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов);
- приобретение и развитие практических умений предусмотренных компетенциями (практические занятия, самостоятельная работа студентов);
- закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе решения конкретных задач на занятиях, выполнения индивидуальных заданий на практических занятиях и их защиты, а так же в процессе сдачи зачета и экзамена.

3 ПОКАЗАТЕЛИ И КРИТЕРИИ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ (РЕЗУЛЬТАТОВ) НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

Сформированность каждой компетенции (или ее части) в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- 1) пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- 2) продвинутый уровень характеризуется превышением минимальных характеристик сформированности компетенций по завершении освоения дисциплины;
- 3) эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

Принимается во внимание наличие и степень сформированности у обучающихся знаний, умений и обладание навыками, которые должны были формироваться в процессе изучения дисциплины.

Уровень освоения компетенций, формируемых дисциплиной:

Описание критериев и шкалы оценивания тестирования:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 85 до 100%
2 балла (продвинутый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 70 до 84%
1 балл (пороговый уровень)	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 50 до 69%
0 баллов	уровень усвоения материала, предусмотренного программой: процент верных ответов на тестовые вопросы от 0 до 49%

Описание критериев и шкалы оценивания теоретического вопроса:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	выставляется студенту, который дал полный ответ на вопрос, показал глубокие систематизированные знания, смог привести примеры, ответил на дополнительные вопросы преподавателя
2 балла (продвинутый уровень)	выставляется студенту, который дал полный ответ на вопрос, но на некоторые дополнительные вопросы преподавателя ответил только с помощью наводящих вопросов
1 балл (пороговый уровень)	выставляется студенту, который дал неполный ответ на вопрос в билете и смог ответить на дополнительные вопросы только с помощью преподавателя
0 баллов	выставляется студенту, который не смог ответить на вопрос

На промежуточную аттестацию (зачет, экзамен) выносятся тест и два теоретических вопроса. Максимально обучающийся может набрать 6 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «отлично», «хорошо», «удовлетворительно» и «неудовлетворительно».

Оценка «отлично» выставляется студенту, который набрал в сумме 6 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра лабораторных работ.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 4 до 5 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра лабораторных работ.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме 3 балла. Обязательным условием является выполнение всех предусмотренных в течение семестра лабораторных работ.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 3 баллов или не выполнил всех предусмотренных в течение семестра лабораторных работ.

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме экзамена

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ПК-3 (ПК-3.1, ПК-3.2)	Способен организовывать разработку системного программного обеспечения ПК-3.1 Планирует разработку системного программного обеспечения ПК-3.2 Организует работу программистов в группе по разработке системного программного обеспечения

Типовые тестовые вопросы:

1. Для каких целей служит вариант использования на диаграмме вариантов использования:

- представляет класс предметной области;
- + описывает действия, совершаемые системой под воздействием актера;
- представляет объект заданного класса;
- описывает события в системе;
- передает сообщение между объектами системы.

2. Диаграмма классов используется для:

- описания функций системы;
- + представления классов системы и статических связей между ними;
- описания взаимодействия системы с внешними объектами;
- задания сервисов системы для актеров;
- описания последовательности событий в системе.

3. Как на диаграмме последовательности отображается время существования объекта в системе:

- функцией отсчета времени;
- фокусом активности;
- + линией жизни;
- временным интервалом между сообщениями;
- типом объекта.

4. Диаграмма кооперации показывает:

- совокупность объектов предметной области;

- + потоки данных между объектами;
- операции объектов;
- атрибуты объектов;
- наследование объектов.

5. Какая модель жизненного цикла подразумевает выполнение проекта без возможности возврата на предыдущие этапы:

- +каскадная;
- с промежуточным контролем;
- спиральная;
- инкрементальная.

6. Какая модель жизненного цикла является итерационной разновидностью каскадной модели:

- каскадная;
- +с промежуточным контролем;
- спиральная;
- инкрементальная.

7. Какая модель жизненного цикла основана на постепенном наращивании функционала с повторными уточнениями задач:

- каскадная;
- с промежуточным контролем;
- +спиральная;
- инкрементальная.

8. Какую модель жизненного цикла предпочтительнее использовать при большом количестве итераций:

- каскадная;
- с промежуточным контролем;
- спиральная;
- +инкрементальная.

9. Диаграмма деятельности предназначена:

- для определения условий перехода между состояниями;
- для описания внутренних функций состояний;
- + для детализации выполняемых системой действий;

для задания времени нахождения в состоянии;
для задания дорожек ответственности классов.

10. Для описания поведения моделируемой системы используется:

диаграмма вариантов использования;
диаграмма компонентов;
диаграмма кооперации;
+ диаграмма состояний;
диаграмма деятельности.

11. Диаграмма компонентов языка UML показывает:

иерархию классов предметной области;
результаты объектной декомпозиции;
+ физическую структуру разрабатываемого программного обеспечения;
функции, выполняемые системой;
потоки данных в системе.

12. Диаграмма развертывания языка UML показывает:

иерархию классов предметной области;
результаты объектной декомпозиции;
аппаратную конфигурацию узлов системы;
+ аппаратную конфигурацию узлов системы с программными компонентами;
функции, выполняемые системой.

13. Архитектура объектно-ориентированной программы представляет собой:

набор процедур и функций;
иерархию классов предметной области;
+ множество объектов, обменивающихся сообщениями;
обработчики внутренних и внешних событий;
множество экранных форм.

14. Класс - это:

+ тип данных, включающий описание свойств и функций объектов;
множество объектов с общим состоянием и поведением;

тип данных, включающий описание функций с возможностью их повторного использования;
множество объектов, имеющих общие свойства и функции.

15. Объект - это:

абстрактный тип данных, включающий состояние и поведение;
набор данных и функций работы с ними;
+экземпляр класса;
тип данных, реализующий функции класса и хранящий его состояние.

16. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствуемой функциональностью называется:

инкапсуляцией;
виртуализацией;
полиморфизмом;
+наследованием.

17. Тестирование белого ящика предназначено:

для проверки правильности выполняемых функций программы;
+ для проверки внутренней логики программы;
для проверки допустимых комбинаций входных данных;
для проверки допустимых комбинаций выходных данных.

18. Как представляется программа при тестировании базового пути:

множеством объектов;
списком операторов;
+ потоковым графом;
списками входных и выходных параметров;
командами языка ассемблера.

19. Количество регионов потокового графа показывает:

+ количество тестовых вариантов в способе тестирования базового пути;
количество циклов в программе;
количество модулей программы;
количество переходов в программе;
количество входных данных программы.

19. Тестирование черного ящика предназначено:

- для проверки допустимых комбинаций входных данных;
- для проверки допустимых комбинаций выходных данных;
- + исследования функции программы на всей области определения;
- определения экстремумов функции;
- проверки точности вычисления функций программы.

20. Технология экстремального программирования предназначена для:

- работы в экстремальных условиях;
- + для работы в условиях неопределенных или быстро меняющихся требований;
- для сокращения объема программного кода;
- для уменьшения времени на разработку технического задания;
- для сокращения количества тестовых вариантов.

21. Основными целями технология экстремального программирования являются:

- повышение качества интерфейса;
- + максимальное сокращение сроков выпуска работоспособной программы;
- минимизация ошибок;
- повышение быстродействия программы;
- оптимизация программного кода.

22. Итерации экстремального программирования выполняются на основе:

- вариантов использования;
- +пользовательских историй;
- анализа исходного текста программы;
- результатов тестирования программы;
- анализа быстродействия программы.

23. Технология визуального программирования это:

- способ создания экранных форм;
- +способ создания программы путём манипулирования графическими объектами вместо написания её текста;
- построение графических образов;
- технология улучшения визуального качества изображений.

Типовые практические задания:

Задание 1

В соответствии с вариантом разработать для заданной предметной области

диаграмму вариантов использования.

Критерии выполнения задания 1

Задание считается выполненным, если обучающийся определил варианты использования создаваемой системы и отношения между ними, а также назначил актеров, определяющих внешние подсистемы по отношению к создаваемой.

Задание 2

В соответствии с вариантом разработать для заданной предметной области диаграмму классов.

Критерии выполнения задания 2

Задание считается выполненным, если обучающийся выделил части предметной области задачи и представил каждую из них соответствующим классом с необходимыми атрибутами и операциями, а также определил отношения в иерархии классов.

Задание 3

В соответствии с вариантом разработать в среде визуального программирования Delphi программу (набор программ) без использования средств ООП.

Критерии выполнения задания 3

Задание считается выполненным, если обучающийся правильно разработал структуру программы, определил все события в программе и написал соответствующие обработчики.

Задание 4

В соответствии с вариантом разработать в среде визуального программирования Delphi программу с использованием средств ООП.

Критерии выполнения задания 4

Задание считается выполненным, если обучающийся правильно определил все объекты предметной области, определил отношения между ними, построил диаграмму классов, выбрал необходимые для решения задачи визуальные компоненты среды Delphi, получил результат..

Задание 5

В соответствии с вариантом разработать и реализовать в среде Delphi диаграмму классов с помощью абстрактных методов.

Критерии выполнения задания 5

Задание считается выполненным, если обучающийся построил и реализовал на Delphi диаграмму классов, используя абстрактные классы.

Задание 6

В соответствии с вариантом разработать и реализовать в среде Delphi диаграмму классов с помощью делегирования.

Критерии выполнения задания 6

Задание считается выполненным, если обучающийся построил и реализовал на Delphi диаграмму классов, используя делегирование как метод построения новых классов из имеющихся.

Типовые теоретические вопросы:

1. Понятие программы, программирования и технологии программирования
2. Методология процедурно-ориентированного программирования
3. Основание, сущность и особенности объектно-ориентированного подхода. Языки объектно-ориентированного программирования.
4. Основные принципы объектно-ориентированного программирования.
5. Функциональное программирование
6. Понятие объекта предметной области. Понятие объектной декомпозиции. Диаграмма объектов.
7. Виды связей между объектами (классами) на диаграммах (ассоциация, включение и обобщение).
8. Объекты и сообщения. Состояние и поведение объектов (чем характеризуются). Операции над объектами.
9. Понятие класса и объекта в ООП. Соответствие между объектами предметной области и программными объектами (классами).
10. Структура определения класса. Конструирование классов.
11. Методы объекта, методы класса и свободные подпрограммы. Статические методы.
12. Ограничение доступа к полям и методам класса: 3 уровня
13. Конструкторы и деструкторы классов.
14. Наследование как средство разработки классов.
15. Понятие статического и динамического связывания программ
16. Разновидности (виды)(реализации) (толкования) полиморфизма
17. Простой (статический) полиморфизм
18. Сложный (истинный) полиморфизм (полиморфные объекты)
19. Динамические виртуальные методы в ООП (Delphi)
20. Виртуальные функции в C++ и Object Pascal (переопределение функций, чистые и пустые виртуальные функции, абстрактные методы и классы).
21. Композиция и наполнение как средство разработки классов
22. Дополнительные средства разработки классов:

- а) Метаклассы
 - б) Делегирование методов. Делегирование в Delphi при обработке событий. Процедурные свойства (property).
 - в) Контейнерные классы
 - г) Параметризованные классы (шаблоны).
23. Обработка исключительных ситуаций в объектно-ориентированных программах.
24. Определение ТППО. Понятие ЖЦПО. Процессы ЖЦ и их содержание
25. Основные этапы разработки ПО или стадии по ГОСТ 19.102-77 «Стадии разработки».
26. Типы (модели) ЖЦ ПО. Стратегии проектирования ПО.
27. Классический ЖЦ (каскадная модель)
28. Макетирование (прототипирование) как процесс создания моделей ПО
29. Инкрементная модель разработки ПО (пошаговая)
30. Спиральная модель разработки ПО
31. Гибкие технологии разработки ПО - XP, Scrum
32. Методология быстрой разработки приложений (RAD – технология разработки ПО (Rapid Application Development))
33. Компонентно–ориентированная модель (технология проектирования и разработки ПО)
34. Модели качества процессов разработки ПО
35. Стандарт SWEBOOK
36. Понятие языка UML.
37. Предметы в UML (структурные, поведения, группирующие, поясняющие).
38. Отношения в UML - зависимость, ассоциация, включение (агрегация и композиция), обобщение, реализация.
39. Диаграммы языка UML:
- а) вариантов использования,
 - б) классов,
 - в) состояний,
 - г) деятельностей,
40. Диаграммы языка UML:
- д) последовательностей,
 - е) кооперации,
 - ж) компонентов,
 - з) размещения
41. Постановка задачи контроля программ.
42. Отладка программ.
43. Этапы и методы тестирования ПО.
44. Тестирование методом «белого ящика» и его особенности
45. Тестирование методом «черного ящика» и его особенности
46. Способы тестирования модулей методом «белого ящика»:
- а) тестирование базового множества путей
 - б) тестирование потоков данных
 - в) тестирование условий

- г) тестирование циклов
- 47. Способы тестирования методом «чёрного ящика»:
 - а) разбиение по классам эквивалентности.
 - б) анализ граничных значений
 - в) анализ причинно-следственных связей.
- 48. Интеграционное тестирование
- 49. Тестирование правильности (системное)
- 50. Оценочное тестирование:
 - а) тестирование восстановления
 - б) тестирование безопасности
 - в) стрессовое тестирование
 - г) тестирование производительности
- 51. Регрессионное тестирование
- 52. Фаззинг
- 53. Понятие и общее описание структур данных и структур хранения
- 54. Классификация структур данных
- 55. Статические структуры данных:
 - а) Векторы (одномерные массивы)
 - б) Массивы (двухмерные)
 - в) Записи
 - г) Таблицы
- 56. Полустатические структуры данных:
 - а) Стек
 - б) Очередь
 - в) Дек (двусторонняя очередь)
- 57. Линейные динамические структуры данных. Односвязные списки (однонаправленные и двунаправленные)
- 58. Нелинейные связные структуры данных:
 - а) Сетевые и древовидные структуры данных
 - б) Бинарные деревья
- 59. Понятие и классификация алгоритмов обработки данных
- 60. Трудоемкость алгоритмов и методы ее оценки.
- 61. Методы оценки ресурсной эффективности алгоритмов
- 62. Базовые алгоритмы обработки данных
- 63. Упорядочивание данных. Алгоритмы внутренней сортировки и их сравнение:
 - а) Сортировка простыми вставками.
 - б) Сортировка простым выбором
 - в) Сортировка простым обменом
 - г) Сортировка методом Шелла
 - д) Метод квадратичной выборки
 - е) Алгоритм “быстрой сортировки”
 - ж) Сортировка методом слияния
- 64. Постановка задачи поиска данных. Классификация алгоритмов поиска. Последовательный поиск.

- 65. Поиск в упорядоченном множестве записей. Бинарный поиск.
- 66. Быстрый поиск без явного упорядочения (сортировки) множества записей:
 - а) Поиск по бинарному дереву
 - б) Использование сбалансированных и оптимальных деревьев поиска. AVL-деревья.
 - в) Сильноветвящиеся деревья и внешний поиск.
- 67. Методы поиска, основанные на использовании неупорядоченных структур данных. Хэш-функции, хэш-адресация и хэш-таблицы.
- 68. Алгоритмы на графах:
 - а) Способы представления графов в ЭВМ.
 - б) Обход вершин графов. Поиск в глубину. Поиск в ширину.
 - в) Задача нахождения пути в графах.
 - г) Построение стягивающего дерева графа.
 - д) Нахождение фундаментального множества циклов в графах

4.2. Курсовая работа (ПК-3.1, ПК-3.2)

Тематика курсовой работы: «Разработка прикладных программ на языке Object Pascal (или C++)». Задание к курсовой работе включает две части.

В рамках первой части задания необходимо разработать объектно-ориентированную (имитационную) модель предметной области и реализовать ее в среде Delphi на языке Object Pascal или в среде Visual Studio на языке C++.

В рамках второй части задания необходимо:

- а) решать задачи на графах (обход графа в ширину и в глубину, поиск путей в графе, поиск кратчайших путей в графе, поиск циклов в графе)
- б) реализовать списковые и древовидные структуры данных (в соответствии с вариантом) и применить их для решения конкретных задач (создание и заполнение списка или дерева, поиск в списке или дереве, вставка или удаление элементов в списке или дереве,
- в) реализовать хэш-таблицы и применить их для решения поставленных задач (выбор хэш-функции, создание и заполнение хэш-таблицы, предотвращение коллизий)

Критерии выполнения курсовой работы

Результаты курсового проектирования оцениваются с учетом:

- 1) качества и полноты выполнения пояснительной записки;
- 2) наличия работающей программы;
- 3) уровня ответов обучающегося на вопросы.

Варианты предметных областей для практических заданий и курсовой работы

Вариант 1

Система поддержки управления библиотекой должна обеспечивать операции (добавление, удаление и изменение) над данными о читателях. В регистрационном списке читателей хранятся следующие сведения: фамилия, имя и отчество читателя; номер его читательского билета и дата выдачи билета. Наряду с регистрационным списком системой должен поддерживаться каталог библиотеки, где хранится информация о книгах: название, список авторов, библиотечный шифр, год и место издания, название издательства, общее количество экземпляров книги в библиотеке и количество экземпляров, доступных в текущий момент. Система обеспечивает добавление, удаление и изменение данных каталога, а также поиск книг в каталоге на основании введенного шифра или названия книги. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждую выданную книгу хранится запись о том, кому и когда была выдана книга, и когда она будет возвращена. При возврате книги в записи делается соответствующая пометка, а сама запись не удаляется из системы. Система должна выдавать следующую справочную информацию:

- какие книги были выданы за данный промежуток времени;
- какие книги были возвращены за данный промежуток времени;
- какие книги находятся у данного читателя;
- имеется ли в наличии некоторая книга.

Вариант задания предусматривает разработку схемы базы данных, хранящей список читателей, каталог книг и записи о выдаче книг.

Вариант 2

Интернет-магазин позволяет делать покупки с доставкой на дом. Клиенты магазина при помощи программы-браузера имеют доступ к каталогу продаваемых товаров, поддержку которого осуществляет Интернет-магазин. В каталоге товары распределены по разделам. О каждом товаре доступна полная информация (название, вес, цена, изображение, дата изготовления и срок годности). Для удобства клиентов предусмотрена система поиска товаров в каталоге. Заполнение каталога информацией происходит автоматически в начале рабочего дня, информация берется из системы автоматизации торговли.

При отборе клиентами товаров поддерживается виртуальная «торговая корзина». Любое наименование товара может быть добавлено в «корзину» или изъято

в любой момент по желанию покупателя с последующим пересчетом общей стоимости покупки. Текущее содержимое «корзины» постоянно показывается клиенту.

По окончании выбора товаров производится оформление заказа и регистрация покупателя. Клиент указывает в регистрационной форме свою фамилию, имя и отчество, адрес доставки заказа и телефон, по которому с ним можно связаться для подтверждения сделанного заказа. Заказы передаются для обработки в систему автоматизации торговли. Проверка наличия товаров на складе и их резервирование Интернет-магазином не производится. Дополнительно требуется разработать схему базы данных, хранящей заказы.

При выполнении этого варианта задания рекомендуем ознакомиться с работой [Коналлен-2001]. Следует определиться, по какому архитектурному шаблону будет строиться Web-приложение («тонкий клиент» или «толстый клиент»). В соответствии с выбранным шаблоном следует построить модели клиентской части магазина и серверной части, промоделировать связи между частями приложения. Для Web-приложений типичными являются следующие *классы*:

- клиентская Web-страница;
- серверная Web-страница (например, CGI-скрипт);
- HTML-форма;
- объект JavaScript.

Дополнительные *связи* между классами Web-приложений:

- link - ссылка с одной страницы на другую;
- build - связь между CGI-скриптом и клиентской страницей, генерируемой при его выполнении;
- submit - связь между формой и серверной Web-страницей, принимающей данные из формы.

Типичные *компоненты*:

- Web-страница (HTML-файл),
- Active Server Page (ASP),
- Java Server Page (JSP),
- сервлет,
- библиотека скриптов (например, подключаемый файл с Javascript-функциями).

Вариант 3

WWW-конференция представляет собой хранилище сообщений в сети Интернет, доступ к которому осуществляется при помощи браузера. Для каждого со-

общения конференции хранятся значения следующих полей: номер сообщения, автор, тема, текст сообщения, дата добавления сообщения, ссылка на родительское сообщение. Начальной страницей конференции является иерархический список сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на сообщения верхнего уровня. Сообщение-ответ всегда имеет ссылку на исходное сообщение. В списке отображаются только темы сообщений, их авторы и даты добавления. Просматривая список, пользователь выбирает сообщение и по гиперссылке открывает страницу с текстом сообщения. Помимо текста на этой странице отображается список (иерархический) сообщений являющихся ответами, ответами на ответы и т.д. Для удобства пользователей необходимо предусмотреть поиск сообщений по автору или по ключевым словам в теме или тексте сообщения.

Сообщения добавляются в конференцию зарегистрированными пользователями, которые при отправке сообщения должны указать своё имя и пароль. Регистрирует новых пользователей модератор конференции - её ведущий. При регистрации пользователь заполняет специальную форму, содержимое которой затем пересылается модератору и запоминается в базе пользователей. Модератор решает, регистрировать пользователя или нет, и отправляет свой ответ.

При добавлении сообщений пользователь имеет возможность начать новую тему или ответить на ранее добавленные сообщения. После добавления сообщения оно доступно для чтения всем пользователям (даже незарегистрированным), и список сообщений обновляется.

Модератор имеет право по тем или иным причинам удалять сообщения любых авторов. Он также может наказывать пользователей, нарушающих правила поведения в конференции, лишая на некоторое время пользователя возможности добавлять и редактировать сообщения.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Наиболее подходящей архитектурой для WWW-конференции является «тонкий клиент», поскольку клиентская часть практически не содержит «бизнес-логики». Единственным её элементом, который может выполняться на стороне клиента, является проверка правильного заполнения полей формы, перед отправкой её содержимого на сервер.

В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация.

Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен пользователям. Пользователям каталога должны быть предоставлены возможности по поиску ресурсов. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать список ресурсов по релевантности (соответствию ключевым словам из запроса) или по дате обновления.

Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая Web-сайты, URL которых хранятся в каталоге.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Как и в варианте «WWW-конференция» самой подходящей архитектурой для каталога является «тонкий клиент», поскольку клиентская часть практически не включает в себя функций «бизнес-логики» кроме проверки содержимого форм перед пересылкой на сервер.

Вариант 5

На экране будильника постоянно отображается текущее время (часы и минуты, например: 12 : 00), двоеточие между числом часов и числом минут загорается и гаснет с интервалом в полсекунды.

Управление будильником осуществляется следующими кнопками:

- кнопкой режима установки времени,
- кнопкой режима установки времени срабатывания,
- двумя отдельными кнопками для установки часов и минут,
- кнопкой сброса сигнала «СБРОС».

На будильнике имеется переключатель режима работы со следующими по-

ложениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР».

Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс. Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени срабатывания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается.

Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению «ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

Вариант 6

Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-жены», «дети-родители» и «братья-сестры». Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. Например, недопустимо, чтобы человек был собственным предком или потомком.

Разработанная модель должна содержать схему базы данных для хранения генеалогических деревьев.

Пользователи системы могут осуществлять поиск полезной информации по дереву:

- находить для указанного члена семьи его детей;
- находить для указанного члена семьи его родителей;
- находить для указанной персоны братьев и сестер, если таковые есть;
- получать список всех предков персоны;
- получать список всех потомков персоны;

- получать список всех родственников персоны;
- прослеживать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр - брат Ольги, Ольга - жена Ивана»).

Вариант 7

В телевизоре имеются следующие устройства: приемник телевизионного сигнала, устройство отображения картинки, память каналов, память настроек, управляющие кнопки, пульт дистанционного управления (ДУ). Управление телевизором осуществляется при помощи кнопок на корпусе (их четыре: «ВКЛ/ВЫКЛ», «-», «+», кнопка начальной установки) и пульта ДУ. Кнопка «ВКЛ/ВЫКЛ» позволяет включать и выключать телевизор. После включения телевизора на экран отображается передача, идущая по каналу №1, при этом используются параметры изображения и значение громкости, сохраненные в памяти настроек.

Память каналов телевизора хранит до 60 каналов. Каналы нумеруются, начиная с нуля. Последовательное переключение каналов осуществляется при помощи кнопок «-» и «+». Нажатие на «+» переключает телевизор на канал с номером, на единицу большим (с 59-го канала телевизор переключается на 0-ой). Нажатие на «-» переключает телевизор на канал с номером, на единицу меньшим (с 0-го канала телевизор переключается на 59-ый).

При нажатии на кнопку начальной установки очищается память каналов телевизора, после чего осуществляется поиск передач и сохранение их частот в памяти каналов. Поиск начинается с нижней границы рабочего диапазона телевизора. На экран телевизора выводится «синий экран». Рабочая частота постепенно увеличивается до тех пор, пока приемник не обнаружит телевизионный сигнал. Найденная передача выводится на экран в течение 10 секунд. Также отображается номер, под которым найденный канал будет сохранен в памяти (начиная с 1).

Составил

к.т.н., доцент кафедры

«Информационная безопасность»

Ю.М. Кузьмин

Оператор ЭДО ООО "Компания "Тензор"

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

ПОДПИСАНО
ЗАВЕДУЮЩИМ
КАФЕДРЫ

ФГБОУ ВО "РГРТУ", РГРТУ, Пржегорлинский Виктор
Николаевич, Преподаватель

08.08.24 05:26 (MSK)

Простая подпись

ПОДПИСАНО
ЗАВЕДУЮЩИМ
ВЫПУСКАЮЩЕЙ
КАФЕДРЫ

ФГБОУ ВО "РГРТУ", РГРТУ, Пржегорлинский Виктор
Николаевич, Преподаватель

08.08.24 05:26 (MSK)

Простая подпись