ПРИЛОЖЕНИЕ 1
к рабочей программе дисциплины

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Кафедра «Электронные вычислительные машины»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

«Технологии программирования»

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль) подготовки
Вычислительные машины, комплексы, системы и сети
Системы автоматизированного проектирования
Системный анализ и инжиниринг информационных процессов

Уровень подготовки

Академический бакалавриат

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная, заочная

Рязань

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 ОПОП.

Цель – оценить соответствие знаний, умений и владений, приобретенных обучающимся в процессе изучения дисциплины, целям и требованиям ОПОП в ходе проведения промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций.

Контроль знаний обучающихся проводится в форме промежуточной аттестации.

Промежуточная аттестация проводится в форме экзамена. Форма проведения экзамена - тестирование, письменный опрос по теоретическим вопросам и выполнение практических заданий.

2 ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

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.

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

Описание критериев и шкалы оценивания практического задания:

Шкала оценивания	Критерий
3 балла (эталонный уровень)	Задача решена верно
2 балла (продвинутый уровень)	Задача решена верно, но имеются неточности в логике решения
1 балл (пороговый уровень)	Задача решена верно, с дополнительными наводящими вопросами преподавателя
0 баллов	Задача не решена

На промежуточную аттестацию в форме экзамена выносится тест, теоретический вопрос и задача. Максимально студент может набрать 9 баллов. Итоговый суммарный балл студента, полученный при прохождении промежуточной аттестации, переводится в традиционную форму по системе «зачтено» и «не зачтено».

Оценка «зачтено» выставляется студенту, который набрал в сумме не менее 3 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «не зачтено» выставляется студенту, который набрал в сумме менее 3 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

3 ПАСПОРТ ОЦЕНОЧНЫХ МАТЕРИАЛОВ ПО ДИСЦИПЛИНЕ

Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Вид, метод, форма оценочного мероприятия
Тема 1. Технология программирования. Основные понятия и подходы	ОПК-8	Экзамен
Тема 2. Создание и управление проектами	ОПК-9	Экзамен
Тема 3. Приемы обеспечения технологичности программных продуктов	ОПК-8, ОПК-9	Экзамен
Тема 4. Определение требований к программному обеспечению и исходных данных для его проектирования	ОПК-8	Экзамен
Тема 5. Анализ требований и определение спецификаций программного обеспечения при структурном подходе	ОПК-8	Экзамен
Тема 6. Проектирование программного обеспечения при структурном подходе	ОПК-8	Экзамен
Тема 7. Анализ требований и определение спецификаций программного обеспечения при объектном подходе	ОПК-8	Экзамен
Тема 8. Проектирование программного обеспечения при объектном подходе	ОПК-8	Экзамен
Тема 9. Тестирование программных продуктов	ОПК-9	Экзамен
Тема 10. Составление программной документации	ОПК-9	Экзамен

4 ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ

4.1. Промежуточная аттестация в форме экзамена/зачета

Код компетенции	Результаты освоения ОПОП Содержание компетенций
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-8.1. Понимает требования к алгоритмам, суть процесса алгоритмизации задач
ОПК-8.2. Выполняет разработку алгоритмического и программного обеспечения для решения прикладных задач

Типовые тестовые вопросы:

1. Традиционным методом организации информационных систем является:
– клиент-клиент;
+ клиент-сервер;
– сервер-сервер;
– размещение всей информации на одном компьютере.

2. Для повышения эффективности разработки программного обеспечения применяют:
– Delphi;
– C++;
+ CASE –средства;
– Pascal.

3. Microsoft.Net является:
+ платформой;
– языком программирования;
– системой управления базами данных;
– прикладной программой.

4. Под CASE – средствами понимают
+ программные средства, поддерживающие процессы создания и сопровождения программного обеспечения;
– языки программирования высокого уровня;
– среды для разработки программного обеспечения;
– прикладные программы.

5. 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.

6. 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.

7. Как на диаграмме последовательности отображается время существования объекта в системе:
- функцией отсчета времени;
- фокусом активности;
+ линией жизни;
- временным интервалом между сообщениями;
- типом объекта.

8. Диаграмма кооперации показывает:
- совокупность объектов предметной области;
+ потоки данных между объектами;
- операции объектов;
- атрибуты объектов;
- наследование объектов.

9. Триггерный переход между состояниями срабатывает:
- при завершении do-деятельности;
- при завершении указанного интервала времени;
+ при наступлении события, внешнего по отношению к исходному состоянию;
- при выполнении заданного условия;
- при совпадении имени внешнего события и внутреннего действия.

10. Нетриггерный переход между состояниями срабатывает:
+ при завершении do-деятельности;
- при завершении указанного интервала времени;
- при наступлении события, внешнего по отношению к исходному состоянию;
- при выполнении заданного условия;
- при совпадении имени внешнего события и внутреннего действия.

11. Какая модель жизненного цикла подразумевает выполнение проекта без возможности возврата на предыдущие этапы:
+каскадная;
с промежуточным контролем;
спиральная;
инкрементальная.

12. Какая модель жизненного цикла является итерационной разновидностью каскадной модели:
каскадная;
+с промежуточным контролем;
спиральная;
инкрементальная.

13. Какая модель жизненного цикла основана на постепенном наращивании функционала с повторными уточнениями задач:
каскадная;
с промежуточным контролем;
+спиральная;
инкрементальная.

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

каскадная;
с промежуточным контролем;
спиральная;
+инкрементальная.

15. 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.

16. Для описания поведения моделируемой системы используется:

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

17. Диаграмма компонентов языка UML показывает:

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

18. 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.

19. 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.

20. 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.

21. 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.

22. Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью называется:
инкапсуляцией;
виртуализацией;
полиморфизмом;
+наследованием.

Типовые практические задания:

Задание 1

В соответствии с вариантом разработать для заданной предметной области диаграмму вариантов использования и описательную спецификацию.

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 для заданной предметной области диаграмму классов и последовательностей.

Критерии выполнения задания 3

Задание считается выполненным, если обучающийся определил сообщения необходимых типов между экземплярами классов системы и представил результаты в графическом виде на языке UML.

Задание 4

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

Критерии выполнения задания 4

Задание считается выполненным, если обучающийся определил потоки данных и представил результаты в графическом виде на языке UML.

Задание 5

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

Критерии выполнения задания 5

Задание считается выполненным, если обучающийся выделил типы программных компонент создаваемой системы, взаимодействие между ними и представил результаты в графическом виде на языке UML.

Задание 6

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

Критерии выполнения задания 6

Задание считается выполненным, если обучающийся выделил узлы, соответствующие аппаратной конфигурации создаваемой системы, каналы их взаимодействия и привязал программные компоненты к соответствующим узлам.

Задание 7

В соответствии с вариантом разработать для заданной предметной области диаграмму состояния для всей системы.

Критерии выполнения задания 7

Задание считается выполненным, если обучающийся выделил состояния создаваемой системы, события, вызывающие переходы между ними и представил результаты в графическом виде на языке UML.

Задание 8

В соответствии с вариантом разработать для заданной предметной области диаграмму деятельности для всей системы.

Критерии выполнения задания 8

Задание считается выполненным, если обучающийся выделил виды деятельности создаваемой системы и их взаимодействия, и представил результаты в графическом виде на языке UML.

Варианты предметных областей для практических заданий 1-8

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 вначале 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 производятся. 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 сообщения 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.

Вариант 4

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 зажигается 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 положениями: «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).

Типовые теоретические вопросы:

1. Понятие жизненного цикла ИС.
2. Жизненный цикл и этапы разработки ПО.
3. Стандартизация процессов жизненного цикла ИС.
4. Виды процессов жизненного цикла.
5. Эволюция моделей жизненного цикла ПО.
6. Эволюция моделей жизненного цикла ПО. Каскадная модель ЖЦ
7. Эволюция моделей жизненного цикла ПО. Модель ЖЦ с промежуточным контролем.
8. Эволюция моделей жизненного цикла ПО. Спиральная модель ЖЦ ПО.
9. Эволюция моделей жизненного цикла ПО. Инкрементальная модель.
10. Выбор процесса жизненного цикла процесса разработки ПО.
11. Построение моделей ИС
12. Классификация моделей разрабатываемых ИС.
13. Язык UML. Диаграммы вариантов использования.
14. Язык UML. Диаграммы классов.
15. Язык UML. Отношения между классами.
16. Язык UML. Диаграммы состояний.
17. Язык UML. Диаграммы деятельности.
18. Язык UML. Диаграммы взаимодействия.
19. Язык UML. Диаграммы компонентов.
20. Язык UML. Диаграммы размещения.
21. Проблемы разработки сложных ИС.
22. Технология экстремального программирования.
23. Технология RUP.
24. Технология визуального программирования.

Код компетенции	**Результаты освоения ОПОП** **Содержание компетенций**
ОПК-9	Способен осваивать методики использования программных средств для решения практических задач

ОПК-9.1. Демонстрирует знания современного состояния информационных технологий и программных средств, применяемых при решении практических задач

ОПК-9.2. Понимает особенности и специфику различных классов программных средств
ОПК-9.3. Осуществляет применение новых методик использования программных средств для решения практических задач

Типовые тестовые вопросы:

1. 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.

2. Как представляется программа при тестировании базового пути:
множеством объектов;
списком операторов;
+ потоковым графом;
списками входных и выходных параметров;
командами языка ассемблера.

3. 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.

4. 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.

5. 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.

6. Основными целями технология экстремального программирования являются:
повышение качества интерфейса;
+ максимальное сокращение сроков выпуска работоспособной программы;
минимизация ошибок;
повышение быстродействия программы;
оптимизация программного кода.

7. Итерации экстремального программирования выполнятся на основе:
вариантов использования;
+пользовательских историй;
анализа исходного текста программы;
результатов тестирования программы;
анализа быстродействия программы.

8. 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.

9. Альфа тестирование проводится:
+ заказчиком в организации разработчика;
- разработчиком в организации заказчика;
- независимыми экспертами;
- после года эксплуатации программы;
- при использовании новой версии операционной системы.

10. Бета тестирование проводится:
+ заказчиком в процессе эксплуатации системы;
- разработчиком в организации заказчика;
- независимыми экспертами;
- после года эксплуатации программы;
- при использовании новой версии операционной системы.

11. Технология быстрой разработки приложений RAD (Rapid Application Development) предназначена для:
+быстрого получения работающих прототипов программной системы;
быстрой компиляции приложений;
для сокращения объема программного кода;
для уменьшения времени на разработку технического задания.

12. Оценка размера приложений по технологии RAD производится на основе:
строк исходного кода;
количества сообщений между объектами;
+функциональных элементов;
количества модулей системы;
количества килобайт исполняемого модуля.

13. Технология Rational Unified Process (RUP) предназначена для:
разработки программных систем для бизнес анализа;
+ создание и сопровождение моделей систем на базе языка UML;
сокращения сроков разработки технического задания;
для сокращения объема программного кода.

14. Для построения моделей частей предметной области задачи используется:
структурная декомпозиция;
+объектная декомпозиция;
пошаговая детализация;
имитационной моделирование.

15. Жизненный цикл программного обеспечения представляет собой:
время, необходимое на кодирование ПО;
время, необходимое на анализ и постановку задачи;

+ период от момента появления идеи разработки ПО до момента прекращения его сопровождения;
период, необходимый для внедрения ПО;
время между появлением новых версий ПО.

16. Какой международный стандарт регламентирует жизненный цикл ИС:
+ISO 12207;
ISO 12107;
ISO 34.101;
ISO 19.102.

17. Цель проекта – это:
а) Сформулированная проблема, с которой придется столкнуться в процессе выполнения проекта
+ б) Утверждение, формулирующее общие результаты, которых хотелось бы добиться в процессе выполнения проекта
в) Комплексная оценка исходных условий и конечного результата по итогам выполнения проекта
г) Описание прибыли, которую можно извлечь по итогам выполнения проекта

18. Проект отличается от процессной деятельности тем, что:
а) Процессы менее продолжительные по времени, чем проекты
б) Для реализации одного типа процессов необходим один-два исполнителя, для реализации проекта требуется множество исполнителей
+ в) Процессы однотипны и цикличны, проект уникален по своей цели и методам реализации, а также имеет четкие срока начала и окончания
г) Среди этих утверждений нет четкого определения отличия

19. Что из перечисленного не относится к «треугольнику проекта», который представляет метафору компромисса между целями и ограничениями проекта?
+ а) Границы
б) Время
в) Качество
г) Стоимость

20. Наибольшее влияние на проект оказывают
а) экологические факторы и инфраструктура
+ б) экономические и правовые факторы
в) культурно-социальные факторы
г) политические и экономические факторы

21. Организационная структура, при которой возможно перераспределение человеческих ресурсов между проектами без реорганизации существующей структуры, называется
+ а) Матричная
б) Функциональная
в) Линейно-функциональная
г) Дивизиональная

Типовые практические задания:

Задание 1. Задана программа на псевдокоде, которую необходимо протестировать с помощью способа тестирования базового пути. При этом следует выполнить следующие шаги:
1. Построить потоковый граф программы.
2. Вычислить цикломатическую сложность тремя разными способами.
3. Определить базовое множество независимых путей графа.
4. Разработать тестовые варианты для каждого пути.

Критерии выполнения задания 1

Задание считается выполненным, если обучающийся правильно построил потоковый граф программы, разработал тестовые варианты для каждого пути из базового множества независимых путей графа.

Варианты для задания 1

Вариант 1

С помощью способа тестирования базового пути разработать тестовые варианты для приведенной ниже программы на псевдокоде:

```
{процедура «Снять деньги со счета через банкомат»}
если (банкомат готов)
 то
  нач
   вставить карту
   ввести PIN-код
   если (время ожидания < 5 минут) и (PIN-код правильный)
    то  нач
          ввести сумму
          если (денег на счете достаточно)
           то выдать деньги
           иначе выдать сообщение «Денег недостаточно»
         кон
    иначе выдать сообщение «Время вышло или PIN-код
                                неправильный»
  кон
```

Вариант 2

```
{процедура «Выполнение операций с банкоматом»}
читать карту
пока (не «Выход из меню»)
 нач
  запросить PIN-код
  если (карта принадлежит банку) и (PIN-код правильный)
   то
    нач
     ввести код операции
     если (операция = «Снять деньги со счета»)
      то выдать требуемую сумму
      иначе
       если (операция = «Просмотреть баланс»)
        то вывести баланс счета
    кон
   иначе выдать сообщение «Карта не принадлежит банку
                  или PIN-код неправильный»
```

```
кон
```

Вариант 3

```
{процедура «Продажа товара через торговый автомат»}
если (торговый аппарат свободен)
 то
  нач
   внести сумму
   выбрать товар
   если (время ожидания не вышло)
    то
     если (товар в наличии) и (денег достаточно)
      то выдать товар
      иначе выдать сообщение «Нет товара или денег
                  недостаточно»
    иначе
     нач
      выдать сообщение «Товар не выбран»
      вернуть деньги
     кон
  кон
```

Вариант 4

```
{процедура «Оплата мобильной связи через банкомат»}
если (банкомат свободен)
 то
  нач
   вставить карту
   ввести номер телефона
   если (время ожидания < 5 минут) и (номер существует)
    то
     нач
      ввести сумму оплаты
      если (денег на карте достаточно)
       то оплатить мобильную связь
       иначе выдать сообщение «Денег недостаточно»
     кон
    иначе выдать сообщение «Время вышло
                  или номер неправильный»
  кон
```

Задание 2. Необходимо протестировать программу с помощью способа разбиения на классы эквивалентности. При этом следует выполнить следующие шаги:
1. По каждому входному параметру определить допустимые и недопустимые классы эквивалентности и представить их в виде таблицы.
2. Построить дерево разбиений.
3. Разработать тестовые варианты для каждого конечного листа дерева.

Критерии выполнения задания 2

Задание считается выполненным, если обучающийся правильно определил допустимые и недопустимые классы эквивалентности по каждому входному параметру и разработал для них тестовые варианты.

Задание 3. Необходимо протестировать программу с помощью способа анализа граничных значений. Для этого требуется:
1. По каждому параметру выявить граничные значения.
2. Разработать тестовые варианты, проверяющие параметры на каждой границе.

Критерии выполнения задания 3

Задание считается выполненным, если обучающийся правильно определил граничные значения по каждому входному параметру и разработал для них тестовые варианты.

Задание 4. Необходимо протестировать программу с помощью способа анализа причинно-следственных связей. Для этого требуется:
1. Выявить причины и следствия.
2. Построить логические выражения, определяющие каждое следствие через причины.
3. Разработать граф причинно-следственных связей.
4. Построить таблицу решений.
5. Для каждого столбца таблицы решений разработать отдельный тестовый вариант.

Критерии выполнения задания 4

Задание считается выполненным, если обучающийся правильно определил причины и следствия, разработал граф причинно-следственных связей, построил таблицу решений и разработал все тестовые варианты.

Варианты для задания 2-4

Вариант 1

Требуется протестировать программу начисления стипендии студентам младших курсов. Средний балл может принимать значения в диапазоне от 12 до 20. Известно, что стипендия определяется согласно следующей таблице:

Курс	Средний балл	Стипендия
1	≥14 <14	Повышенная Обычная
2	≥16 <16	Повышенная Обычная
3	≥18 <18	Повышенная Обычная

Вариант 2

Требуется протестировать программу оценивания знаний студентов 1, 2, 3-го курса по дисциплине «История». Известно, что результат определяется согласно следующей таблице:

Курс	% решенных заданий	Результат
1	[90-100] [0-90)	Удовлетворительно Неудовлетворительно
2	[50-100] [0-50)	Удовлетворительно Неудовлетворительно
3	[30-100] [0-30)	Удовлетворительно Неудовлетворительно

Вариант 3

Требуется протестировать программу начисления премии сотрудникам банка в зависимости от стажа работы и должности. Стаж может принимать значения в диапазоне от 1 до 25.

Должность	Стаж	Премия
Администрация	≥5 <5	20 % 10 %
Менеджеры	≥10 <10	20 % 10 %
Операторы	≥20 <20	20 % 10 %

Вариант 4

Требуется протестировать программу расчета транспортного налога в зависимости от типа автомобиля и мощности двигателя. Известно, что результат определяется согласно следующей таблице:

Тип машины	Мощность двигателя, л.с.	Налог, руб.
Легковая	≥100 <100	5000 1000
Грузовая	≥150 <150	10000 5000

Задание 5

В соответствии с вариантом задания необходимо: определить даты проекта, выбрать тип календаря выполнения проекта, определить календарь рабочей недели, составить список основных задач проект, определить параметры задач.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 5

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, при этом верно определил даты проекта, правильно выбрал тип календаря выполнения проекта и определил календарь рабочей недели, перечислил все требуемые задачи проекта.

Задание 6

В соответствии с вариантом задания необходимо: задать все основные сведения о проекте, определить список основных задач и составляющих их подзадач, требуемых для успешного выполнения проекта.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 6

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, при этом верно определил все сведения о проекте, правильно перечислил основные задачи проекта и выделил подзадачи.

Задание 7

В соответствии с вариантом задания необходимо: задать все основные сведения о проекте, определить список и параметры задач всех уровней, требуемых для успешного выполнения проекта, структурировать список задач, задать вехи.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 7

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, верно определил все основные сведения о проекте, список и параметры задач всех уровней, требуемых для успешного выполнения проекта, правильно структурировал список задач и задал вехи.

Задание 8

В соответствии с вариантом задания необходимо: задать пул ресурсов, определить конкретные материальные и трудовые ресурсы, требуемые для успешного завершения проекта, задать ручное выравнивание ресурсов.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 8

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, правильно определил пул ресурсов, конкретные материальные и трудовые ресурсы, требуемые для успешного завершения проекта, задал ручное выравнивание ресурсов.

Задание 9

В соответствии с вариантом задания необходимо: выполнить назначение ресурсов задачам проекта; определить стандартные ставки для ресурсов проекта, определить стоимость каждого из ресурсов; определить стоимости отдельных задач и проекта в целом.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 9

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, правильно выполнил назначение ресурсов задачам проекта; верно определил стандартные ставки для ресурсов проекта, стоимость каждого из ресурсов и стоимости отдельных задач и проекта в целом.

Задание 10

В соответствии с вариантом задания необходимо: определить стандартные ставки для ресурсов проекта, определить стоимость каждого из ресурсов; определить стоимости отдельных задач и проекта в целом; выполнить анализ и выравнивание загрузки ресурсов.

Задание должно быть реализовано с помощью программного средства Microsoft Project.

Критерии выполнения задания 10

Задание считается выполненным, если обучающийся разработал файл проекта с расширением *.mpp, используя программу Microsoft Project, правильно определил стандартные ставки для ресурсов проекта, стоимость каждого из ресурсов, стоимости отдельных задач и проекта в целом, выполнил анализ и выравнивание загрузки ресурсов.

Варианты предметных областей для заданий 5-10.

Вариант 1. *Разработка системы для учета заграничных командировок в государственной компании с высоким уровнем секретности*

Государственной кампании с высоким уровнем секретности требуется разработать систему для учета заграничных командировок, в которые часто отправляются её

сотрудники. Система должна обладать повышенным уровнем безопасности, а также должна быть полностью защищена от утечки информации.

Доступ к системе будет иметь определенная группа лиц, работающих в этой кампании.

В системе должны ежедневно обновляться данные о командировках сотрудников: их длительности, месте, цели и конкретных адресах, которые были посещены за время пребывания в другой стране, а также другая более подробная информация о поездке.

При необходимости кампания будет отправлять данные о командировках другим организациям, но только при наличии специального запроса и после прохождения всех проверок.

Система создается на длительное время, поэтому должна быть рассчитана на большие объемы информации.

Государственная кампания планирует нанимать специалистов для внедрения и сопровождения системы.

Система должна быть разработана в течение двух лет.

Вариант 2. Разработка системы для управления курьерской доставкой товаров

Служба курьерской доставки товаров открыла новые офисы в нескольких городах России и планирует расширять зоны доставки как малогабаритных, так и крупногабаритных грузов в пределах страны.

Для этого требуется мощная система управления данной службой. Необходимо вести учет товара на складах одновременно в нескольких городах, контролировать перемещение товаров по стране, вести прием заказов и подсчет стоимости доставки на конкретное расстояние. С помощью этой системы также планируется начисление заработной платы сотрудникам, мониторинг состояния транспортных средств компании и т.д.

Служба курьерской доставки планирует запуск этой системы не позднее, чем через 8 месяцев.

Сопровождение системы не требуется.

Вариант 3. Разработка системы для учета товаров на складе небольшой фирмы

Недавно появившаяся фирма арендовала складское помещение для хранения товаров.

Количество товаров ежедневно меняется, происходит отгрузка имеющихся и поступление новых партий товаров от различных фирм.

В один день может произойти и отгрузка и поступление одновременно.

Требуется система для учета товаров на складе: даты их закупки и продажи, их количества и цены.

Причем, если запасы какого-либо товара на складе близки к нулю, система должна информировать пользователя об этом.

Фирма молодая и не обладает крупным капиталом для реализации системы, потому её представители готовы обсудить детали и возможные планы сокращения стоимости проекта. Максимальный срок ожидания системы - 6 месяцев.

Сопровождение не требуется.

Вариант 4. Разработка системы для проведения рекламных компаний

Компания проводит акции, обзванивая потенциальных клиентов, проживающих в одном городе.

Требуется разработать систему автоматического набора телефонных номеров. Система должна соединять оператора с клиентами. Каждому сотруднику должна предоставляться учетная запись для входа в систему. Должна поддерживаться

одновременная работа нескольких операторов. Во время проведения каждой акции клиентам должны присваиваться различные статусы для слежения за ходом кампании. Система должна взаимодействовать с существующей в компании системой учета клиентов.

Срок разработки системы – 1 год. Планируется ввести продукт в рабочий цикл компании в два этапа: сначала часть с базой клиентов и подсистему статистики, затем – автоматизации работы и контроля сотрудников.

Типовые теоретические вопросы:

1. Тестирование ПО. Функциональное тестирование.
2. Тестирование ПО. Структурное тестирования.
3. Тестирование ПО. Способ тестирования базового пути.
4. Тестирование ПО. Способ разбиения на классы эквивалентности.
5. Тестирование ПО. Способ анализа граничных значений.
6. Тестирование ПО. Способ анализа причинно следственных связей.
7. Организация процесса тестирования.
8. Методика тестирования ПО, разработанного на основе процедурного подхода. Тестирование элементов. Алгоритм работы тестового драйвера.
9. Методика тестирования ПО, разработанного на основе процедурного подхода. Алгоритм работы тестового драйвера.
10. Методика тестирования ПО, разработанного на основе процедурного подхода. Нисходящее тестирование интеграции.
11. Методика тестирования ПО, разработанного на основе процедурного подхода. Восходящее тестирование интеграции.
12. Тестирование правильности ПО. Основные компоненты конфигурации программной системы. Альфа-тестирование.
13. Тестирование правильности ПО. Основные компоненты конфигурации программной системы. Бета-тестирование.
14. Системное тестирование.
15. Методика тестирования ПО, разработанного на основе объектно-ориентированного подхода.

Оператор ЭДО ООО "Компания "Тензор"
ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
СОГЛАСОВАНО **ФГБОУ ВО "РГРТУ", РГРТУ,** Костров Борис Васильевич, Заведующий кафедрой ЭВМ **14.10.26** 10:23 (MSK) Простая подпись