

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

КАФЕДРА «ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

«Объектно-ориентированное программирование»

Направление подготовки

38.03.05 «Бизнес информатика»

ОПОП академического бакалавриата

«Бизнес информатика»

Квалификация (степень) выпускника — бакалавр

Форма обучения — очная, заочная

1 ОБЩИЕ ПОЛОЖЕНИЯ

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

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной профессиональной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности общекультурных, общепрофессиональных и профессиональных компетенций, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины (модуля), организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретенных обучающимися в ходе выполнения индивидуальных заданий на практических занятиях и лабораторных работах. При оценивании результатов освоения практических занятий и лабораторных работ применяется шкала оценки «зачтено – не зачтено». Количество лабораторных и практических работ и их тематика определена рабочей программой дисциплины, утвержденной заведующим кафедрой.

Результат выполнения каждого индивидуального задания должен соответствовать всем критериям оценки в соответствии с компетенциями, установленными для заданного раздела дисциплины.

Промежуточный контроль по дисциплине осуществляется проведением экзамена и теоретического зачета.

Форма проведения экзамена – письменный ответ по утвержденным экзаменационным билетам, сформулированным с учетом содержания учебной дисциплины. В экзаменационный билет включается два теоретических вопроса и одна задача. После выполнения письменной работы обучающегося производится ее оценка преподавателем и, при необходимости, проводится теоретическая беседа с обучаемым для уточнения экзаменационной оценки.

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 выносятся тест, два теоретических вопроса и 2 задачи. Максимально студент может набрать 15 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 15 баллов (выполнил все задания на эталонном уровне). Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «хорошо» выставляется студенту, который набрал в сумме от 10 до 14 баллов при условии выполнения всех заданий на уровне не ниже продвинутого. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «удовлетворительно» выставляется студенту, который набрал в сумме от 5 до 9 баллов при условии выполнения всех заданий на уровне не ниже порогового. Обязательным условием является выполнение всех предусмотренных в течение семестра практических заданий.

Оценка «неудовлетворительно» выставляется студенту, который набрал в сумме менее 5 баллов или не выполнил всех предусмотренных в течение семестра практических заданий.

ПК-4.3 Программирует ИС

Типовые тестовые вопросы

Правильный ответ помечен знаком +

1. Свойство, при котором объекты содержат описание атрибутов и действий одновременно:

- а) Наследование
- б) Полиморфизм
- + в) Инкапсуляция

2. Один из принципов объектно-ориентированного программирования:

- + а) Инкапсуляция
- б) Ингаляция
- в) Инструкция

3. Один из принципов объектно-ориентированного программирования:

- а) Отдача
- б) Передача
- + в) Наследование

4. Один из принципов объектно-ориентированного программирования:

- а) Абстракционизм
- + б) Полиморфизм
- в) Монорфизм

5. Метод определения объектов, при котором производные объекты наследуют свойства от своих потомков:

- а) Монорфизм
- б) Полиморфизм
- + в) Наследование

6. Свойство объектов, при котором действие с одинаковыми именами вызывает различное поведение для различных объектов:

- + а) Полиморфизм
- б) Передача
- в) Монорфизм

7. Данные, характеризующие состояние объекта:

- а) Доли объекта
- б) Части объекта
- + в) Атрибуты объекта

8. Под объектами понимают:

- а) Всю абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
- + б) Некоторую абстрактную сущность, заданную набором имен атрибутов и имен методов поведения
- в) Некоторую видимую сущность, заданную набором имен атрибутов и имен методов поведения

9. Совокупность объектов, характеризующаяся общностью методов и свойств:

- + а) Класс
- б) Вид
- в) Род

10. Изменение состояния объекта в ответ на какое-либо действие:

- а) Значение
- + б) Событие
- в) Данность

11. Действие, которое может выполнить объект:

- + а) Метод
- б) Событие
- в) Свойство

12. Характеристика объекта:

- а) Событие
- б) Данность
- + в) Свойство

13. Совокупность свойств и методов:

- + а) Объект
- б) Свойство
- в) Событие

14. Тип, соответствующий классу:

- а) Объективный тип
- + б) Объектный тип
- в) Видимый тип

15. Компоненты, которые видны во время работы приложения, с ними напрямую может взаимодействовать пользователь, называются:

- а) Абстрактными
- б) Видимыми
- + в) Визуальными

16. Все объекты в Delphi происходят от объекта:

- а) TObject
- + б) TObject [-]
- в) VObject

17. Свойство Align отвечает за:

- а) Уменьшение компонента на форме
- б) Увеличение компонента на форме
- + в) Выравнивание компонента на форме

18 Свойство Caption отвечает за:

- а) Назначение компонента на форме
- + б) Название компонента на форме
- в) Вид компонента на форме

19. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам:

- а) Статическая
- + б) Дружеская
- в) Шаблонная

20. Какая из перечисленных функций не может быть конструктором:

- а) String (const int a)
- б) String (String & s)
- + в) void String ()

21. Свойство Name отвечает за:

- а) Название компонента
- + б) Имя компонента
- в) Назначение компонента

22. Имя формы, используется для управления формой и доступа к компонентам формы:

- + а) Свойство формы Name
- б) Значение формы Name
- в) Следствие формы Name

23. Расстояние от верхней границы формы до верхней границы экрана:

- а) Значение формы Top
- б) Следствие формы Top
- + в) Свойство формы Top

24. Этот компонент предназначен для вывода текста на поверхность формы:

- + а) Label
- б) Edit
- в) Button

25. Компонент, представляющий собой поле ввода-редактирования строки символов:

- а) Memo
- + б) Edit
- в) CheckBox

Вопросы с ответами

1. Что такое ООП?

Объектно-ориентированное программирование или ООП – это методология программирования, которая представляет программное обеспечение в виде набора объектов. Каждый объект – это не что иное, как экземпляр класса.

2. Разница между процедурным программированием и ООП?

Процедурное программирование:

1. Основано на функциях.
2. Определяет данные для всей программы.
3. В нем нет возможности повторного использования кода.
4. Следует концепции нисходящего программирования.
5. Природа языка сложна.
6. Трудно изменять, расширять и поддерживать код.

Объектно-ориентированное программирование:

1. Основано на реальных объектах.
2. Инкапсулирует данные.
3. Обеспечивает больше возможностей для повторного использования кода.
4. Следует парадигме программирования «снизу вверх».
5. Он менее сложен по своей природе, поэтому его легче модифицировать, расширять и поддерживать.

3. Зачем использовать ООП?

ООП позволяет повторно использовать код. Данные и код связаны вместе с помощью инкапсуляции. ООП имеет возможности для сокрытия данных, поэтому частные данные могут храниться и сохранять конфиденциальность. Задачи могут быть разделены на различные части, что упрощает их решение. Концепция полиморфизма обеспечивает гибкость, так как одна сущность может иметь несколько форм.

4. Каковы основные концепции ООП?

Основными концепциями ООП являются:

1. Наследование
2. Инкапсуляция

3. Полиморфизм

4. Абстракция

5. Что такое инкапсуляция?

Инкапсуляция является частью концепции ООП. Она относится к объединению данных с методами, которые работают с этими данными. Это также помогает ограничить любой прямой доступ к некоторым компонентам объекта.

6. Что такое абстракция?

Абстракция – это концепция ООП для построения структуры объектов реального мира. Она «показывает» только существенные атрибуты и «прячет» ненужную информацию от посторонних глаз. Основная цель абстракции – скрыть ненужные детали от пользователей. Это одна из самых важных концепций ООП.

7. Что такое перегрузка методов?

Существует концепция, согласно которой два или более метода могут иметь одинаковое имя. Но они должны иметь разные параметры, разное количество параметров, разные типы параметров или и то, и другое. Такие методы известны как перегруженные методы, и эта особенность называется перегрузкой методов.

8. Что такое переопределение методов?

Переопределение методов – это концепция объектно-ориентированного программирования. Это особенность языка, которая позволяет подклассу или дочернему классу предоставлять конкретную реализацию метода, который уже предоставлен одним из его суперклассов или родительских классов.

9. Типы наследования в ООП

1. Гибридное наследование
2. Множественное наследование
3. Одноуровневое наследование
4. Многоуровневое наследование
5. Иерархическое наследование

10. Что такое объект?

Объект – это экземпляр класса, а также он обладает собственной индивидуальностью и поведением.

11. Что такое метод?

Метод – это описание набора инструкций, который также называется процедурой.

12. Что такое класс?

Класс – это тип данных, определяемый пользователем, который содержит переменные, свойства и методы. Он также помогает найти свойства объекта.

13. Что такое конструкторы?

Конструктор имеет то же имя, что и класс. Конструктор также является особым видом метода. Он используется для инициализации объектов класса.

14. Типы конструкторов

Типы конструкторов зависят от языка:

1. Частный конструктор.
2. Конструктор по умолчанию.
3. Конструктор копирования.
4. Статический конструктор.
5. Параметризованный конструктор.

15. В чем разница между классом и структурой?

Класс – это определенный пользователем чертеж, на основе которого создаются объекты. Он состоит из методов (набор инструкций), которые выполняются над объектами.

Структура – это определенная пользователем коллекция переменных. Структуры также являются различными типами данных.

16. Что такое модификаторы доступа?

Модификаторы доступа или спецификаторы доступа – это ключевые слова в объектно-ориентированных языках. Они помогают установить доступность классов, методов и других членов.

17. Какие языки относятся к концепции ООП?

Simula известен как первый объектно-ориентированный язык программирования, наиболее популярными языками ООП являются:

- Java.
- JavaScript.
- Python.
- C++.
- Visual Basic . NET.
- Ruby.
- Scala.
- PHP.

18. Что такое наследование?

Если вы порождаете класс от другого класса, это называется наследованием.

Дочерний класс наследует все публичные и защищенные свойства и методы от родительского класса. Дочерний класс также может иметь свои собственные свойства и методы. Наследуемый класс определяется с помощью ключевого слова extends.

Множественное наследование:

1. Если класс наследует более одного базового класса.
2. Пример: класс, объясняющий ребенка. Этот класс – ребенок – наследует от двух базовых классов, которыми являются мать и отец.

Многоуровневое наследование:

1. Если класс наследуется от другого класса, который сам является подклассом какого-либо другого базового класса.
2. Пример: класс под названием спортивный велосипед, который наследуется от базового класса велосипед. В свою очередь, класс велосипед наследуется от другого класса транспортное средство.

19. Что такое гибридное наследование?

Сочетание множественного и многоуровневого наследования известно как гибридное наследование.

20. Что такое иерархическое наследование?

Когда один базовый класс имеет более одного подкласса. Например, класс фруктов может иметь в качестве подклассов «яблоко», «манго» и т. д.

21. Каковы ограничения наследования?

Эти ограничения увеличивают время выполнения и трудозатраты и требуют переходов от одного класса к другому. Родительский класс и дочерний класс всегда тесно связаны. Внесение изменений в программу потребует изменений как в родительском, так и в дочернем классе.

Наследование требует тщательной реализации, иначе это приведет к неправильным результатам.

22. Что такое суперкласс?

Суперкласс или базовый класс – это класс, который работает как родитель для некоторых других классов.

Например, класс транспортного средства является суперклассом класса велосипеда.

23. Что такое подкласс?

Подкласс – это класс, который наследуется от другого класса. Например, класс велосипед является подклассом или производным от класса транспортное средство.

24. Что такое полиморфизм?

Полиморфизм является одним из наиболее используемых и основных понятий в языках ООП.

Полиморфизм – концепция, согласно которой различные классы могут использоваться с одним и

тем же интерфейсом. Каждый из этих классов может иметь свою собственную реализацию интерфейса.

25. Что такое статический полиморфизм?

Статический полиморфизм или статическое связывание – это один из видов полиморфизма, который возникает во время компиляции. Примером полиморфизма во время компиляции является: перегрузка методов.

26. Что такое динамический полиморфизм?

Динамический полиморфизм, динамическое связывание или полиморфизм во время выполнения – это также часть полиморфизма, который в основном реализуется во время выполнения программы. Пример полиморфизма во время выполнения программы: переопределение метода.

27. Что такое перегрузка операторов?

Перегрузка операторов используется для реализации операторов, использующих определенные пользователем типы, на основе передаваемых вместе с ними аргументов.

28. Проведите различие между перегрузкой и переопределением.

При перегрузке два или более методов имеют одинаковое имя, но у них разные параметры или сигнатура.

Переопределение – дочерний класс наследует методы с теми же параметрами/подписью, которые присутствуют в базовом классе.

29. Зачем нужна инкапсуляция?

Инкапсуляция позволяет скрыть данные и обернуть данные и код, который работает над ними, в единое целое.

30. В чем разница между модификаторами доступа public, private и protected?

Модификатор	Доступность из самого класса	Доступность из производного класса	Доступность из любого места программы
Public	Да	Да	Да
Private	Да	Нет	Нет
Protected	Да	Да	Нет

31. Что такое абстракция данных?

Абстракция данных – одна из самых важных особенностей ООП. Она позволяет отображать только важную информацию. Она помогает скрыть детали реализации.

Например, при использовании мобильного телефона вы знаете, как можно отправить сообщение или позвонить кому-то, но не знаете, как это происходит на самом деле.

Это и есть абстракция данных, поскольку детали реализации скрыты от пользователя.

32. Как достичь абстракции данных?

Абстракция данных может быть достигнута двумя способами:

1. Абстрактный класс.
2. Абстрактный метод.

33. Что такое абстрактный класс?

Абстрактный класс – это класс, который состоит из абстрактных методов.

Что же такое абстрактный метод?

Эти методы в основном объявлены, но не определены, и если эти методы должны быть использованы позже в каком-то подклассе, то эти методы должны быть определены исключительно в подклассе.

34. Проведите различие между абстракцией данных и инкапсуляцией

Абстракция:

1. Решает проблему на уровне проектирования.
2. Помогает скрыть детали реализации.

Инкапсуляция:

1. Решает проблему на уровне реализации.
2. Обертывает код и данные в единое целое и помогает скрыть их от посторонних глаз.

35. Что такое виртуальные функции?

Виртуальные функции – это часть функций, которые присутствуют в родительском классе, и они переопределяются подклассом.

Эти функции помогают достичь полиморфизма во время выполнения.

36. Что такое деструктор?

Деструктор – это метод, который вызывается автоматически при уничтожении объекта.

Деструктор также восстанавливает пространство кучи, которое было выделено для уничтоженного объекта. Он также начинает закрывать файлы и соединения базы данных объекта и т. д.

37. Что такое конструктор копирования?

Конструктор копирования в основном создает объекты путем копирования переменных из другого объекта того же класса. Основной задачей конструктора копирования является создание нового объекта из существующего.

38. Для чего используется Finalize?

Finalize используется для освобождения неуправляемых ресурсов, а также для очистки перед сборкой мусора (GC). Он выполняет задачи управления памятью.

39. Что такое сборка мусора (GC)?

Сборка мусора – это часть автоматического управления памятью. Сборщик мусора помогает освободить места, занятые объектами. Эти места перестают существовать.

40. Что такое конечная переменная?

Конечная переменная не изменяется и всегда ссылается на один и тот же объект благодаря свойству необратимости.

41. Что такое исключение?

Исключение – это вид сообщения, которое прерывает и появляется, когда возникает проблема с нормальным выполнением программы. Исключения выдают ошибку и передают эту ошибку обработчику исключений для ее устранения. Состояние программы сохраняется, как только возникает исключение.

42. Что такое обработка исключений?

Обработка исключений в объектно-ориентированном программировании является наиболее важной концепцией. Она используется для управления ошибками. Обработчик исключений помогает выбрасывать ошибки, а затем перехватывать их для дальнейшего решения.

43. В чем разница между ошибкой и исключением?

Ошибка: в основном относится к проблеме, и эти проблемы не должны возникать в приложениях.

Исключение: в основном это условие, которое приложение может попытаться выполнить.

44. Что такое блок try/catch?

Блок try/catch помогает обрабатывать исключения. Блок try объясняет набор утверждений, в которых может возникнуть ошибка. Блок catch в основном перехватывает исключение.

45. Что такое блок finally?

Блок finally выполняется, когда блок try завершается, а также выполняется даже в случае возникновения неожиданного исключения. Блок finally обычно содержит какую-то важную часть программы.

46. Можете ли вы вызвать метод базового класса, не создавая его экземпляр?

Да, вы можете вызывать базовый класс без его инстанцирования, но есть некоторые условия, которые необходимо соблюдать:

1. Метод должен быть статическим.
2. Базовый класс наследуется каким-либо другим подклассом.

1. Понятие «объектно-ориентированный подход».
2. Определение класса экземпляра класса.
3. Что такое абстрактные типы данных.
4. Структура сложных систем.
5. В чем заключается декомпозиция сложных систем.
6. Определение сложности программного обеспечения.
7. Пять признаков сложной системы.
8. Алгоритмическая декомпозиция.
9. Объектно-ориентированная декомпозиция.
10. Сравнение алгоритмической и объектно-ориентированной композиции.
11. Роль абстракции в ООП.
12. Роль иерархии в ООП.
13. Интерфейс класса, что это и зачем нужен.
14. В чем смысл скрытия реализации (что это такое).
15. Что такое конструктор и деструктор класса.
16. Что такое постоянство объектов.
17. Что такое прототип объекта.
18. Что такое «обертывание кода».
19. Наследование (для чего применяется).
20. Что такое композиция.
21. Что такое инкапсуляция
22. Что такое полиморфизм
23. Что такое модульность системы
24. Что такое фреймворк
25. Поясните сущность иерархии
26. Для чего необходим контроль типов
27. Параллелизм в системе, объясните, что это.
28. Что является объектом, что не является объектом (примеры)
29. Состояние системы поясните термин
30. Поведение объекта, что это.
31. Роли и обязанности
32. Индивидуальность
33. Примеры объектно ориентированных языков
34. XML и переносимые данные.
35. В чем кроется необходимость переносимых данных
36. UML, для чего применяется
37. Принцип реализации класса в языках программирования
38. Что такое указатель
39. Что такое интерпретатор
40. Что такое компилятор

Составил
доцент кафедры ЭВМ
к.т.н., доцент

Е.Р. Муратов

Заведующий кафедрой ЭВМ,
д.т.н., профессор

Б.В. Костров