

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Малышев Александр Васильевич  
Должность: Заведующий кафедрой  
Дата подписания: 16.06.2023 12:54:54  
Уникальный программный ключ:  
с44с65fc5eb466e5e378с4db413465be7586с86f

## МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:  
Заведующий кафедрой  
программной инженерии

  
А.В. Малышев  
(подпись, инициалы, фамилия)

«17» июня 2022 г.

ОЦЕНОЧНЫЕ СРЕДСТВА  
для текущего контроля успеваемости  
и промежуточной аттестации обучающихся  
по дисциплине

Объектно-ориентированный анализ и программирование  
(наименование дисциплины)

ОПОП ВО 02.03.03 Математическое обеспечение и администрирование  
информационных систем  
код и наименование ОПОП ВО

# 1. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

## 1.1. ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ

### Тема 1. Основы языка C#

1. Исторический экскурс в создание C#.
2. Байт-код CIL (Common Intermediate Language).
3. Common Language Runtime.
4. Основные синтаксические конструкции языка.
5. Структура программы на C#. Пространство имен.
6. Директива using.
7. Метод Main().
8. Параметры программы.
9. Технология создания приложения в среде Microsoft Visual Studio Community

### Тема 2. Объектно-ориентированный подход к моделированию предметной области

1. Моделирование предметной области как необходимый этап разработки программных продуктов.
2. Основные понятия объектно-ориентированного моделирования (объект, класс, взаимоотношения объектов и классов).
3. Основы графического языка моделирования UML
4. Разновидности UML-диаграмм

### Тема 3. Введение в классы C#

1. Основы объектно-ориентированного подхода к программированию. 2. Определение класса. Синтаксис объявления класса.
3. Создание объекта класса.
4. Обращение к полям и методам объекта.
5. Области видимости.
6. Жизненный цикл объекта.
7. Конструкторы. Конструктор по умолчанию. Конструктор с параметрами.
8. Использование инициализаторов.
9. Деструкторы и уборка мусора. Garbage Collector.
10. Объекты в качестве параметров методов.
11. Передача параметров по значению и по ссылке.
12. Передача объектов по значению.
13. Объекты в качестве возвращаемых значений методов.

### Тема 4. Наследование и полиморфизм

1. Простое наследование.
2. Конструкторы и наследование.
3. Неявный вызов конструктора базового класса.
4. Явный вызов конструктора базового класса.
5. Деструкторы и наследование. Наследование и сокрытие имен.

6. Ключевое слово base.
7. Совместимость типов объектов.
8. Полиморфизм.
9. Виртуальные и переопределенные методы.

Тема 5. Абстрактные классы и интерфейсы, абстрактные свойства.

1. Абстрактные классы
2. абстрактные методы.
3. Наследование абстрактных классов и переопределение абстрактных методов.
4. Интерфейсы.
5. Классы, реализующие интерфейса
6. Наследование интерфейсов.
7. Свойства классов, их объявление и наследование.
8. Абстрактные свойства.

Тема 6. Статические члены класса и индексы

1. Статические члены класса.
2. Статические классы.
3. Индексы: объявление и реализация.
4. Работа с индексами.
5. Особенности индексов.
6. Перегрузка методов и операторов.
7. Перегрузка методов.
8. Методы с переменным числом параметров.
9. Ключевое слово `param`.
10. Перегрузка операторов.
11. Операторы, подлежащие и не подлежащие перегрузке.

Тема 6. Тестирование объектно-ориентированных программ

1. Методы тестирования объектно-ориентированного программного обеспечения.
2. Тестирование отдельных модулей (unit-тестирование).
3. Разработка тестовых наборов и тестирующих программ.
4. Интеграционное тестирование

**Шкала оценивания:** 10-балльная.

**Критерии оценивания:**

Каждый вопрос (задание) в тестовой форме оценивается по дихотомической шкале: выполнено – 1 балл, не выполнено – 0 баллов.

Применяется следующая шкала перевода баллов в оценку по 5-балльной шкале:

- 9 -10 баллов соответствуют оценке «отлично»;
- 7 -8 баллов – оценке «хорошо»;
- 6 -7 баллов – оценке «удовлетворительно»;
- 5 баллов и менее – оценке «неудовлетворительно».

## 2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

### 2.1. ТЕМЫ КУРСОВЫХ РАБОТ (КУРСОВЫХ ПРОЕКТОВ) (при наличии курсовых

1. Программа для моделирования поведения небесных тел в поле взаимного тяготения
2. Программа для моделирования работы противопожарной сигнализации
3. Программа для моделирования работы сервиса по продаже железнодорожных билетов
4. Приложение для моделирования работы системы отопления умного дома
5. Программа для моделирования работы банкомата
6. Программа для демонстрации принципов функционирования лазерного черно-белого принтера
7. Программа для моделирования работы умного холодильника
8. Программа для демонстрации принципов функционирования мультиварки
9. Приложение для управления данными для магазина товаров для дома
10. Приложение для управления данными о регистрации транспортных средств
11. Программная реализация игры с пошаговым выполнением инструкций
12. Программа для создания графических изображений с применением кисти
13. Программа для создания словарей
14. Программа для моделирования работы зоопарка
15. Программная реализация игры жанра Tower Defense
16. Программа для моделирования работы автомата для продажи напитков
17. Программа для моделирования работы предприятия общественного питания
18. Программа моделирования работы автосервисной мастерской
19. Программа для моделирования работы музыкального плеера
20. Программа для симуляции боевых сражений
21. Программная реализация образовательной среды университета
22. Программа для регистрации участников соревнований
23. Программная реализация логической игры с элементами стратегии
24. Программа для учета студенческих групп
25. Программа для учета данных об автомобильных товарах
26. Программа моделирования работы бытового прибора на примере микроволновой печи
27. Программа складского учета
28. Клиент-серверное приложение для обмена графическими

сообщениями в режиме реального времени

29. Программа для хранения и отображения данных о спортивных товарах
30. Программа для учета и визуализации движения городского транспорта
31. Программа распознавания лица и глаз человека
32. Программа для управления работой ресторана
33. Программа для симуляции футбольных соревнований
34. Программная реализация игры с построением защитных сооружений
35. Программа генерации снарядов в игре жанра Scrolling shooters

*Шкала оценивания курсовых работ (или курсовых проектов): 100-балльная.*

*Критерии оценивания (нижеследующие критерии оценки являются примерными и могут корректироваться):*

**85-100 баллов** (или оценка «отлично») выставляется обучающемуся, если тема курсовой работы раскрыта полно и глубоко, при этом убедительно и аргументированно изложена собственная позиция автора по рассматриваемому вопросу; курсовая работа демонстрирует способность автора к сопоставлению, анализу и обобщению; структура курсовой работы четкая и логичная; изучено большое количество актуальных источников, включая дополнительные источники, корректно сделаны ссылки на источники; самостоятельно подобраны убедительные примеры; основные положения доказаны; сделан обоснованный и убедительный вывод; сформулированы мотивированные рекомендации; выполнены требования к оформлению курсовой работы.

**70-84 баллов** (или оценка «хорошо») выставляется обучающемуся, если тема курсовой работы раскрыта, сделана попытка самостоятельного осмысления темы; структура курсовой работы логична; изучены основные источники, правильно оформлены ссылки на источники; приведены уместные примеры; основные положения и вывод носят доказательный характер; сделаны рекомендации; имеются незначительные погрешности в содержании и (или) оформлении курсовой работы.

**50-69 баллов** (или оценка «удовлетворительно») выставляется обучающемуся, если тема курсовой работы раскрыта неполно и (или) в изложении темы имеются недочеты и ошибки; отмечаются отступления от рекомендованной структуры курсовой работы; количество изученных источников менее рекомендуемого, сделаны ссылки на источники; приведены самые общие примеры или недостаточное их количество; вывод сделан, но имеет признаки неполноты и неточности; рекомендации носят формальный характер; имеются недочеты в содержании и (или) оформлении курсовой работы.

**Менее 50 баллов (или оценка «неудовлетворительно»)** выставляется обучающемуся, если тема курсовой работы не раскрыта и (или) в изложении темы имеются грубые ошибки; структура курсовой работы нечеткая или не определяется вообще; количество изученных источников значительно менее рекомендуемого, неправильно сделаны ссылки на источники или они отсутствуют; не приведены примеры или приведены неверные примеры; отсутствует вывод или автор испытывает затруднения с выводами; не соблюдаются требования к оформлению курсовой работы.

## 2.2. БАНК ВОПРОСОВ И ЗАДАНИЙ В ТЕСТОВОЙ ФОРМЕ

1. Выберите наиболее подходящее определение Класса

- Тип, содержащий набор функций
- Тип, который отображает состояние некоторого объекта
- Тип, описывающий поведение некоторой сущности
- Тип, описывающий характеристики и поведение объекта

2. Что НЕ является важной частью определения ООП?

- Обязательно использование абстрактных классов;
- Объектно-ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы
- Каждый объект является экземпляром определенного класса
- Классы образуют иерархии

3. Тип данных, описывающий на языке программирования модель некоторой сущности реального мира или абстрактного понятия это -...

- Класс
- Объект
- Пространство имён
- Директива

4. Возможность создавать новые определения классов на основе существующих - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

5. Поддержка выполнения нужного действия в зависимости от типа передаваемого объекта - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

6. Способность скрывать детали реализации объектов от пользователей этих объектов - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

7. Сущность в адресном пространстве вычислительной системы, появляющаяся при создании переменной типа класса это - ...

- Класс
- Объект
- Пространство имён
- Директива

8. Что НЕ является важной частью определения ООП?

- Объектно-ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы;
- Каждый объект является экземпляром определенного класса;
- Классы образуют иерархии
- Обязательно использование абстрактных классов

9. . Установите соответствие понятий и определений

- |   |   |
|---|---|
| <input type="checkbox"/> Запись таблицы базы данных | Совокупность данных в одной строке таблицы базы данных  |
| <input type="checkbox"/> Поле таблицы базы данных   | Совокупность данных в одном столбце таблицы базы данных |
| <input type="checkbox"/> Структура записи таблицы   | Совокупность имен столбцов и типов данных               |
| <input type="checkbox"/> Файл базы данных           | Совокупность Структуры записи таблицы и данных          |

10. Определите последовательность действий для создания запроса :

- выбрать таблицу
- выбрать столбцы
- задать условие отбора
- выполнить запрос

11. Укажите правильную последовательность действий в жизненном цикле объекта

- Создается объект класса

Ответ 1

- Вызывается конструктор для объекта

Ответ 2

- Конструктор выделяет память под объект

Ответ 3

- Программа работает с объектом

Ответ 4

- Объект выходит из области видимости или удаляется

Ответ 5

- Вызывается деструктор для освобождения памяти.

Ответ 6

12. Метод для считывания значения свойства

- Get
- Set
- Take
- Give

13. Метод для записи значения свойства

- Get
- Set
- Take
- Give

14. Метод, который вызывается при разрушении объекта класса (когда не осталось активных ссылок на данный объект).

- Деструктор
- Конструктор
- Инициализатор
- Нет такого метода

15. Инициализатор - это...

- Способ вызвать из текущего конструктора другой конструктор данного класса или конструктор базового класса;
- Создание, активация, подготовка к работе, определение параметров;
- Приведение программы или устройства в состояние готовности к использованию
- Программа, которая выделяет память под объект.

16. Как сделать свойство класса доступным только для записи?

- Прописать в свойстве только get;



- Прописать в свойстве только set
- Сделать данный член класса protected
- Сделать данный член класса private

17. Как сделать свойство класса доступным только для записи?

- Прописать в свойстве только get;
- Прописать в свойстве только set
- Сделать данный член класса protected
- Сделать данный член класса private

18. Обязательное требование для конструктора

- Имя совпадает с именем класса;
- Не имеет параметров
- Записывается в начале объявления класса
- Должен возвращать void

19. Метод, который инициализирует объект при создании

- Конструктор
- Деструктор
- Объявление
- Проектировщик

20. UML Диаграмма классов:

- частный случай диаграммы деятельности
- частный случай диаграммы прецедентов
- соответствует статическому виду системы
- служит для моделирования структуры классов системы и связей между ними

21. Актор – это:

- внешняя сущность по отношению к компьютерной системе, которая может только снабжать информацией систему
- внешняя сущность по отношению к компьютерной системе, которая может только получать информацию из системы
- внешняя сущность по отношению к компьютерной системе, которая взаимодействует с этой системой
- внутренняя сущность компьютерной системы, которая может как получать информацию из системы, так и снабжать информацией систему

22. В языке UML интерфейс – это:

- совокупность ролей и других элементов, которые, работая совместно, производят некоторый совместный эффект, не сводящийся к простой сумме слагаемых;
- описание последовательности выполняемых системой действий, которая производит наблюдаемый результат, значимый для какого-то определенного актора
- совокупность операций, которые определяют сервис (набор услуг), предоставляемый классом или компонентом
- это физическая заменяемая часть системы, которая соответствует некоторому набору классов и обеспечивает его реализацию

23. Для моделирования поведения системы в языке UML могут использоваться следующие диаграммы:

- диаграмма вариантов использования
- диаграмма развёртывания
- диаграмма компонентов
- диаграмма деятельности
- диаграмма последовательности

24. Класс на UML диаграмме содержит следующие поля

- имя класса
- атрибуты класса
- операции класса
- входные данные
- выходные данные

25. К основным структурным сущностям языка UML можно отнести следующие:

- классы
- интерфейсы
- автоматы
- прецеденты
- состояния
- компоненты

26. Между вариантами использования на диаграмме вариантов использования могут существовать следующие связи:

- зависимость

- использование
- расширение
- ассоциация

27. Словарь языка UML включает следующие строительные блоки:

- отношения
- диаграммы
- аннотации
- классы
- сущности
- интерфейсы

28. Язык UML был разработан для того, чтобы:

- моделировать системы целиком, от концепции до исполняемого файла, с помощью объектно-ориентированных методов;
- создать такой язык моделирования, который может использоваться не только людьми, но и компьютерами;
- объединить уже существующие языки визуального моделирования как OMG, CORBA, ORG
- решить проблему масштабируемости, которая присуща сложным системам, предназначенным для выполнения ответственных задач;

29. Драконы умеют летать (как, например, птицы) и ползать (как, например, ящерицы). С точки зрения ООП, примером чего является данная ситуация (выберите наиболее точный вариант)?

- Инкапсуляция
- Композиция
- Полиморфизм
- Множественное наследование

30. От какого системного класса наследуют неявно наследуют все классы в языке C#?

- object
- Class
- Console
- Program

31. Класс, находящийся на вершине иерархии наследования называют:

- Верхним
- Базовым
- Абстрактным
- Виртуальным

32. Какое максимальное количество наследников может быть у класса?

- 1
- 2
- 5
- сколько угодно

33. Может ли наследник класса иметь наследника?

- Может
- Не может
- Может, но только если его базовый класс - абстрактный;
- Может, но только если его базовый класс — виртуальный.

34. Класс, наследующий поля и методы называют:

- Потомком
- Сыном
- Предком
- Наследующим

35. К проявлению какого элемента ООП можно отнести следующую фразу:

"Перемещая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

Инкапсуляция

- Полиморфизм

Наследование

36. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов?

37. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов?

- Модификаторы доступа
- Виртуальные методы
- Обработка исключений

- Статические методы

38. \_\_\_\_\_ в ООП языках обычно позволяют обеспечить инкапсуляцию объектов

39. Фразу: "Перемещая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

К проявлению какого элемента ООП можно отнести \_\_\_\_\_

40. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов \_\_\_\_\_

41. Как выглядит блок catch?

- catch(<тип ошибки>){//тело}
- public catch(<параметры>){//тело}
- catch{//тело}
- catch this

42. Какой вид исключения является наиболее общим?

- Exception
- ParseException
- Exclusion
- ApplicationException

43. Какой комбинации блоков обработки исключений нет?

- catch- finally
- try-catch;
- try-finally
- try-catch- finally

44. Как рекомендуется располагать блоки catch и try и почему?

- Сначала наиболее специализированные, потом наиболее общие, чтобы можно было узнать какая именно ошибка произошла;
- Рекомендаций нет
- Сначала наиболее общие, потом наиболее специализированные, чтобы можно было проследить иерархию ошибки;
- Сначала наиболее частые, потом наиболее редкие, чтобы обработка исключения происходила быстрее

45. Сколько блоков catch может иметь один блок try?

- 1
- 2
- 3
- Много

46. Что обязательно должно быть в блоке try?

- Параметры;
- Модификатор доступа;
- Тело;
- Всё перечисленное.

47. Что помещается в блок catch?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

48. Что помещается в блок finally?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

Код, который может сгенерировать ошибку; помещается в блок \_\_\_\_\_

49. Что помещается в блок try?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

50. Код, создающий пользовательское исключение, помещается в блок \_\_\_\_\_

51. Код, обрабатывающий ошибку, помещается в блок \_\_\_\_\_

1. Выберите наиболее подходящее определение Класса

- Тип, содержащий набор функций
- Тип, который отображает состояние некоторого объекта
- Тип, описывающий поведение некоторой сущности
- Тип, описывающий характеристики и поведение объекта

52. Что НЕ является важной частью определения ООП?

- Обязательно использование абстрактных классов;
- Объектно-ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы
- Каждый объект является экземпляром определенного класса
- Классы образуют иерархии

53. Тип данных, описывающий на языке программирования модель некоторой сущности реального мира или абстрактного понятия это -...

- Класс
- Объект
- Пространство имён
- Директива

54. Возможность создавать новые определения классов на основе существующих - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

55. Поддержка выполнения нужного действия в зависимости от типа передаваемого объекта - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

56. Способность скрывать детали реализации объектов от пользователей этих объектов - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

57. Сущность в адресном пространстве вычислительной системы, появляющаяся при создании переменной типа класса это -...

- Класс
- Объект
- Пространство имён

- Директива

58. Что НЕ является важной частью определения ООП?

- Объектно-ориентированное программирование использует в качестве основных логических конструктивных элементов объекты, а не алгоритмы;
- Каждый объект является экземпляром определенного класса;
- Классы образуют иерархии
- Обязательно использование абстрактных классов

59. . Установите соответствие понятий и определений

- |   |   |
|---|---|
| <input type="checkbox"/> Запись таблицы базы данных | Совокупность данных в одной строке таблицы базы данных  |
| <input type="checkbox"/> Поле таблицы базы данных   | Совокупность данных в одном столбце таблицы базы данных |
| <input type="checkbox"/> Структура записи таблицы   | Совокупность имен столбцов и типов данных               |
| <input type="checkbox"/> Файл базы данных           | Совокупность Структуры записи таблицы и данных          |

60. Определите последовательность действий для создания запроса :

- выбрать таблицу
- выбрать столбцы
- задать условие отбора
- выполнить запрос

61. Укажите правильную последовательность действий в жизненном цикле объекта

- Создается объект класса

Ответ 1

- Вызывается конструктор для объекта

Ответ 2

- Конструктор выделяет память под объект

Ответ 3

- Программа работает с объектом

Ответ 4

- Объект выходит из области видимости или удаляется

Ответ 5

- Вызывается деструктор для освобождения памяти.



Ответ 6

62. Метод для считывания значения свойства

- Get
- Set
- Take
- Give

63. Метод для записи значения свойства

- Get
- Set
- Take
- Give

64. Метод, который вызывается при разрушении объекта класса (когда не осталось активных ссылок на данный объект).

- Деструктор
- Конструктор
- Инициализатор
- Нет такого метода

65. Инициализатор - это...

- Способ вызвать из текущего конструктора другой конструктор данного класса или конструктор базового класса;
- Создание, активация, подготовка к работе, определение параметров;
- Приведение программы или устройства в состояние готовности к использованию
- Программа, которая выделяет память под объект.

66. Как сделать свойство класса доступным только для записи?

- Прописать в свойстве только `get`;
- Прописать в свойстве только `set`
- Сделать данный член класса `protected`
- Сделать данный член класса `private`

67. Как сделать свойство класса доступным только для записи?

- Прописать в свойстве только `get`;

- Прописать в свойстве только set
- Сделать данный член класса protected
- Сделать данный член класса private

#### 68. Обязательное требование для конструктора

- Имя совпадает с именем класса;
- Не имеет параметров
- Записывается в начале объявления класса
- Должен возвращать void

#### 69. Метод, который инициализирует объект при создании

- Конструктор
- Деструктор
- Объявление
- Проектировщик

#### 70. UML Диаграмма классов:

- частный случай диаграммы деятельности
- частный случай диаграммы прецедентов
- соответствует статическому виду системы
- служит для моделирования структуры классов системы и связей между ними

#### 21. Актор – это:

- внешняя сущность по отношению к компьютерной системе, которая может только снабжать информацией систему
- внешняя сущность по отношению к компьютерной системе, которая может только получать информацию из системы
- внешняя сущность по отношению к компьютерной системе, которая взаимодействует с этой системой
- внутренняя сущность компьютерной системы, которая может как получать информацию из системы, так и снабжать информацией систему

#### 72. В языке UML интерфейс – это:

- совокупность ролей и других элементов, которые, работая совместно, производят некоторый совместный эффект, не сводящийся к простой сумме слагаемых;
- описание последовательности выполняемых системой действий, которая производит наблюдаемый результат, значимый для какого-то определенного актора

- совокупность операций, которые определяют сервис (набор услуг), предоставляемый классом или компонентом
- это физическая заменяемая часть системы, которая соответствует некоторому набору классов и обеспечивает его реализацию

73. Для моделирования поведения системы в языке UML могут использоваться следующие диаграммы:

- диаграмма вариантов использования
- диаграмма развёртывания
- диаграмма компонентов
- диаграмма деятельности
- диаграмма последовательности

74. Класс на UML диаграмме содержит следующие поля

- имя класса
- атрибуты класса
- операции класса
- входные данные
- выходные данные

75. К основным структурным сущностям языка UML можно отнести следующие:

- классы
- интерфейсы
- автоматы
- прецеденты
- состояния
- компоненты

76. Между вариантами использования на диаграмме вариантов использования могут существовать следующие связи:

- зависимость
- использование
- расширение
- ассоциация

77. Словарь языка UML включает следующие строительные блоки:

- отношения

- диаграммы
- аннотации
- классы
- сущности
- интерфейсы

78. Язык UML был разработан для того, чтобы:

- моделировать системы целиком, от концепции до исполняемого файла, с помощью объектно-ориентированных методов;
- создать такой язык моделирования, который может использоваться не только людьми, но и компьютерами;
- объединить уже существующие языки визуального моделирования как OMG, CORBA, ORG
- решить проблему масштабируемости, которая присуща сложным системам, предназначенным для выполнения ответственных задач;

79. Драконы умеют летать (как, например, птицы) и ползать (как, например, ящерицы). С точки зрения ООП, примером чего является данная ситуация (выберите наиболее точный вариант)?

- Инкапсуляция
- Композиция
- Полиморфизм
- Множественное наследование

80. От какого системного класса наследуют неявно наследуют все классы в языке C#?

- object
- Class
- Console
- Program

81. Класс, находящийся на вершине иерархии наследования называют:

- Верхним
- Базовым
- Абстрактным
- Виртуальным

82. Какое максимальное количество наследников может быть у класса?

- 1
- 2
- 5
- сколько угодно

83. Может ли наследник класса иметь наследника?

- Может
- Не может
- Может, но только если его базовый класс - абстрактный;
- Может, но только если его базовый класс — виртуальный.

84. Класс, наследующий поля и методы называют:

- Потомком
- Сыном
- Предком
- Наследующим

85. К проявлению какого элемента ООП можно отнести следующую фразу:

"Перемещая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

Инкапсуляция

- Полиморфизм
- Наследование

86. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов?

87. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов?

- Модификаторы доступа
- Виртуальные методы
- Обработка исключений
- Статические методы

88. \_\_\_\_\_ в ООП языках обычно позволяют обеспечить инкапсуляцию объектов

89. Фразу: "Перемещая рукоятку коробки передач автомобиля, человек может не задумываться о самом механизме переключения. С точки зрения водителей все коробки передач работают одинаково, хотя их механизмы на самом деле могут отличаться".

К проявлению какого элемента ООП можно отнести \_\_\_\_\_

90. Какие механизмы в ООП языках обычно позволяют обеспечить инкапсуляцию объектов \_\_\_\_\_

91. Как выглядит блок catch?

- catch(<тип ошибки>){//тело}
- public catch(<параметры>){//тело}
- catch{//тело}
- catch this

92. Какой вид исключения является наиболее общим?

- Exception
- ParseException
- Exclusion
- ApplicationException

93. Какой комбинации блоков обработки исключений нет?

- catch- finally
- try-catch;
- try-finally
- try-catch- finally

95. Как рекомендуется располагать блоки catch и try и почему?

- Сначала наиболее специализированные, потом наиболее общие, чтобы можно было узнать какая именно ошибка произошла;
- Рекомендаций нет
- Сначала наиболее общие, потом наиболее специализированные, чтобы можно было проследить иерархию ошибки;
- Сначала наиболее частые, потом наиболее редкие, чтобы обработка исключения происходила быстрее

96. Сколько блоков catch может иметь один блок try?

- 1
- 2
- 3

- Много

97. Что обязательно должно быть в блоке try?

- Параметры;
- Модификатор доступа;
- Тело;
- Всё перечисленное.

98. Что помещается в блок catch?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

99. Что помещается в блок finally?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

Код, который может сгенерировать ошибку; помещается в блок \_\_\_\_\_

100. Что помещается в блок try?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;
- Код, который выполняется всегда;
- Код, создающий пользовательское исключение.

101. Код, создающий пользовательское исключение, помещается в блок \_\_\_\_\_

102. Код, обрабатывающий ошибку, помещается в блок \_\_\_\_\_

**Шкала оценивания результатов тестирования:** в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по очно-заочной и заочной формам обучения – 60 баллов (установлено положением П 02.016).

Максимальный балл за тестирование представляет собой разность двух чисел: максимального балла по промежуточной аттестации для данной формы обучения (36 или 60) максимального балла за решение компетентностно-ориентированной задачи (6).

Балл, полученный обучающимся за тестирование, суммируется с баллом,

выставленным ему за решение компетентностно-ориентированной задачи.

Общий балл по промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по шкале (*указать нужное: по 5-балльной шкале или дихотомической шкале*) следующим образом (*привести одну из двух нижеследующих таблиц*):

Соответствие 100-балльной и 5-балльной шкал

| <i>Сумма баллов по 100-балльной шкале</i> | <i>Оценка по 5-балльной шкале</i> |
|---|-----------------------------------|
| 100-85                                    | отлично                           |
| 84-70                                     | хорошо                            |
| 69-50                                     | удовлетворительно                 |
| 49 и менее                                | неудовлетворительно               |

***Критерии оценивания результатов тестирования:***

Каждый вопрос (задание) в тестовой форме оценивается по дихотомической шкале: выполнено – **2 балла**, не выполнено – **0 баллов**.



### 3. КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ЗАДАЧИ

1. Дана таблицы базы данных.

**Шкала оценивания решения компетентностно-ориентированной задачи:** в соответствии с действующей в университете балльно-рейтинговой системой оценивание результатов промежуточной аттестации обучающихся осуществляется в рамках 100-балльной шкалы, при этом максимальный балл по промежуточной аттестации обучающихся по очной форме обучения составляет 36 баллов, по очно-заочной и заочной формам обучения – 60 (установлено положением П 02.016).

Максимальное количество баллов за решение компетентностно-ориентированной задачи – 6 баллов.

Балл, полученный обучающимся за решение компетентностно-ориентированной задачи, суммируется с баллом, выставленным ему по результатам тестирования.

Общий балл промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по шкале (*указать нужно: по 5-балльной шкале или дихотомической шкале*) следующим образом (*привести одну из двух нижеследующих таблиц*):

Соответствие 100-балльной и 5-балльной шкал

| <i>Сумма баллов по 100-балльной шкале</i> | <i>Оценка по 5-балльной шкале</i> |
|---|-----------------------------------|
| 100-85                                    | отлично                           |
| 84-70                                     | хорошо                            |
| 69-50                                     | удовлетворительно                 |
| 49 и менее                                | неудовлетворительно               |