

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Малышев Александр Васильевич
Должность: Заведующий кафедрой
Дата подписания: 01.10.2024 10:01:51
Уникальный программный ключ:
с44с65fc5eb466e5e378с401174698e7586с86f

МИНОБРАЗОВАНИЯ И НАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:

Заведующий кафедрой

Программной инженерии

(наименование кафедры полностью)



А.В. Малышев

(подпись)

«8» мая 2024 г.

ОЦЕНОЧНЫЕ СРЕДСТВА

для текущего контроля успеваемости
и промежуточной аттестации обучающихся
по дисциплине

Объектно-ориентированный анализ и программирование

(наименование дисциплины)

02.03.03 Математическое обеспечение и администрирование
информационных систем, направленность (профиль) «Интеллектуальный
анализ данных в экономике»

(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.

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 интерфейс
6. Наследование интерфейсов.
7. Свойства классов, их объявление и наследование.
8. Абстрактные свойства.

Тема 6. Статические члены класса и индексаторы 1.

Статические члены класса.

2. Статические классы.
3. Индексаторы: объявление и реализация.
4. Работа с индексаторами.
5. Особенности индексаторов.
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

Критерии оценки:

7-8 баллов (или оценка «отлично») выставляется обучающемуся, если он демонстрирует глубокое знание содержания вопроса; дает точные определения основных понятий; аргументированно и логически стройно излагает учебный материал;

иллюстрирует свой ответ актуальными примерами (типовыми и нестандартными), в том числе самостоятельно найденными; не нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

5-6 баллов (или оценка «хорошо») выставляется обучающемуся, если он владеет содержанием вопроса, но допускает некоторые недочеты при ответе; допускает незначительные неточности при определении основных понятий; недостаточно аргументированно и (или) логически стройно излагает учебный материал; иллюстрирует свой ответ типовыми примерами.

1-4 баллов (или оценка «удовлетворительно») выставляется обучающемуся, если он освоил основные положения контролируемой темы, но недостаточно четко дает определение основных понятий и дефиниций; затрудняется при ответах на дополнительные вопросы; приводит недостаточное количество примеров для иллюстрирования своего ответа; нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

0 баллов (или оценка «неудовлетворительно») выставляется обучающемуся, если он не владеет содержанием вопроса или допускает грубые ошибки; затрудняется дать основные определения; не может привести или приводит неправильные примеры; не отвечает на уточняющие и (или) дополнительные вопросы преподавателя или допускает при ответе на них грубые ошибки.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

2.1. ТЕМЫ КУРСОВЫХ РАБОТ (КУРСОВЫХ ПРОЕКТОВ)

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 издавать швые определения классы 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. Сущность в адресном пространстве вычислительной системы, появляющаяся при создании переменной типа класса это - ...

- Класс
- Объект
- Пространство имён
- Директива

10. Метод для считывания значения свойства

- Get
- Set
- Take
- Give

11. Метод для записи значения свойства

- Get
- Set
- Take
- Give

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

- Деструктор
- Конструктор
- Инициализатор
- Нет такого метода

13. 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.
14. 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
15. 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
16. Обязательное требование для конструктора
- Имя совпадает с именем класса;
 - Не имеет параметров
 - Записывается в начале объявления класса
 - Должен возвращать void
17. Метод, который инициализирует объект при создании
- Конструктор
 - Деструктор
 - Объявление
 - Проектировщик
18. 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
19. 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

20. 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

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

диаграмма вариантов использования диаграмма развёртывания диаграмма компонентов диаграмма деятельности диаграмма последовательности

22. Класс на UML диаграмме содержит следующие поля ^мя класса

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

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

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

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

- зависимость
- использование
- расширение

- ассоциация

25. Словарь языка UML включает следующие строительные блоки:

- отношения
- диаграмма
- аннотации
- классы
- сущности
- интерфейсы

26. Язык UML был разработан для того, чтобы:

- оделировать 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, TORBA, ORG
 - решить проблему масштабируемости, которая присуща сложным системам, предназначенным для выполнения ответственных задач;

27. 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

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

- object
- Class
- Console
- Program

29. Класс, находящийся на вершине иерархии наследования называют:

- Верхним
- Базовым
- Абстрактным

- Виртуальным

30. Какое максимальное количество наследников может быть у класса?

- 0
 - 1
- сколько угодно

31. 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.

32. Класс, наследующий поля и методы называют:

- Потомком
- Сыном
- Предком
- Наследующим

33. 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

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

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

35. Как выглядит блок catch?

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

36. Какой вид исключения является наиболее общим?

- Exception
- ParseException
- Exdusion
- ApplicationException

37. Какой комбинации блоков обработки исключений нет?

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

38. 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

39. Сколько блоков catch может иметь один блок try?

- 1
- 2
- 3
- Много

40. Что обязательно должно быть в блоке try?

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

41. Что помещается в блок catch?

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

42. Что помещается в блок finally?

- Код, который может сгенерировать ошибку;
- Код, обрабатывающий ошибку;

- Код, который выполняется всегда;
 - Код, создающий пользовательское исключение.
43. Что помещается в блок try?
- Код, который может сгенерировать ошибку;
 - Код, обрабатывающий ошибку;
Код, который выполняется всегда;
 - Код, создающий пользовательское исключение.
44. Что помещается в блок try?
- Код, который может сгенерировать ошибку;
 - Код, обрабатывающий ошибку;
Код, который выполняется всегда;
 - Код, создающий пользовательское исключение.
45. Код, создающий пользовательское исключение, помещается в блок
- 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
46. 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
47. Тип данных, описывающий на языке программирования модель некоторой сущности реального мира или абстрактного понятия это -...
- Класс
 - Объект
 - Пространство имён
 - Директива
48. Возможность издавать швые определения класшв на основе существующих - это
- полиморфизм
 - наследование
 - инкапсуляция
 - абстракция
49. Поддержка выполнения нужного действия в зависимости от типа

передаваемого объекта - это

- полиморфизм
- наследование
- инкапсуляция
- абстракция

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

- полиморфизм
- наследование
- инкапсуляция
- абстракция

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

- Класс
- Объект
- Пространство имён

52. 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

53. 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

54. 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

55. Обязательное требование для конструктора

- Имя совпадает с именем класса;

- не имеет параметров
 - Записывается в начале объявления класса
 - Должен возвращать void
56. Метод, который инициализирует объект при создании
- Конструктор
 - Деструктор
 - Объявление
 - Проектировщик
57. 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

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 02.018).

Максимальный балл за тестирование представляет собой разность двух чисел: максимального балла по промежуточной аттестации для данной формы обучения (36) и 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 5-балльной шкале следующим образом:

Соответствие 100-балльной и 5-балльной шкал

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

2 Вопросы в открытой форме

- 1) Верно ли, что в программе, написанной на языке C#, надо описывать все используемые в ней переменные?
- 2) Какие существуют варианты описания целых типов на языке C#?
- 3) Какие существуют варианты описания вещественных типов на языке C#?
- 4) Каким образом описать на языке C# переменную символьного типа?
- 5) Каким образом описать на языке C# л переменную огического типа?
- 6) Запишите на языке C# следующие числа:
 1. $-25,8 \cdot 10^{-7}$; 10^6 ; $0,5 \cdot 10^6$; 7,48; 2; 4/1000.
- 7) Запишите следующие числа без десятичного порядка:
 1. -0.00027E+4; 759E-3; 1E1.
- 8) Вычислите значения выражений: $a \parallel b \ \&\& \ !a$ при $a=true$ и $b=false$; $t \ \&\& \ (p \%3==0)$ при $t=true$, $p=1011$;
 - i. $(x*y) \ \&\& \ (y>x)$ при $x=2$, $y=1$;
 - ii. $(x*y!=0) \parallel (y>x)$ при $x=2$, $y=1$; $!(a \ \&\& \ b)$ при $a=false$, $b=true$.
- 9) Запишите на C# выражение, истинное при выполнении данного условия:
 - i. x принадлежит отрезку $[0,1]$; x лежит вне отрезка $[0,1]$; x принадлежит отрезку $[2,5]$ или $[-1,1]$; x лежит вне отрезка $[2,5]$ или $[-1,1]$.
- 10) Поясните структуру и правила выполнения условных операторов.
- 11) Запишите указанное действие в виде одного условного оператора:
 \cos^2x , при $0 < x < 2$; $y =$
 - i. $1 - \begin{cases} \sin^2x, & \text{иначе.} \\ \end{cases}$
- 12) Какое значение будет иметь переменная z после выполнения операторов?
 - i. `int z=0, x=1, y=□1;`
 - ii. `if (x>0) if (y>0) z=1; else z=2;`
- 13) Какое значение будет иметь переменная z после выполнения операторов?
 - i. `int z=0, x=□1, y=1; if (x) { if (y>0) z=1;} else z=2;`
- 14) Какое значение будет иметь переменная z после выполнения операторов?
 - i. `int z=0, x=0, y=1;`
 - ii. `if (x) { if (y>0) z=1;} else z=2;`
- 15) Какое значение будет иметь переменная z после выполнения операторов?
 - i. `int z=0, x=3, y=1; if (x) { if (y) z=1;} else z=2;`
- 16) Укажите ошибки в следующих операторах:
 - i. `if (1<x<2) x=x+1; y:=0; else x=0; y+= 1;`
- 17) Какое значение будет иметь переменная z после выполнения операторов?
 - i. `int z=0, x=0, y=1;`

- ii. `if (!x) { if (!(y-1)) z=1;} else z=2;`
- 18) Какое значение будет иметь переменная *z* после выполнения операторов:
- i. `int z=0, x=1, y=1; if (!x) { if (!(y-1)) z=1;} else z=2;`
- 19) Найдите ошибки во фрагменте программы:
- i. `{int n,x; switch (k) case + : x:=x-4 break; case '-','*': x=5;`
- ii. `}`
- iii. Какие правила нарушены и как можно исправить ошибки?
- 20) Какие типы можно использовать для выражения–селектора и констант в операторе `switch`?
- 21) Как устанавливается соответствие между `if` и `else` при вложениях условных операторов?
- 22) Может ли массив содержать один элемент?
- 23) Можно ли во время выполнения программы изменить размер массива?
- 24) Могут ли элементами некоторого массива быть числа 1,
- 25) 1.41, 4.98, 30?
- 26) Верно ли, что тип элементов массива может быть любым?
- 27) Может ли типом индекса массива быть тип *int* или *float*?
- 28) Имеются описания:
1. `typedef float a[6]; typedef double c[10];`
2. `const b[]={“среда”, “четверг”, “пятница”}; a mas1={3,6,7,9,1,0}; c mas2;`
3. `int d[7];`
- ii. `char s[]={“ABC”};` Какие массивы определены?
- iii. Какие значения могут принимать массивы?
- 29) Как указать первый и последний элементы каждого из описанных массивов?
- 30) Допустимы ли в среде **C# Builder** операции над массивами как единым целым?
- 31) Дан фрагмент программы:
- 32) `char x[40]; char y[39];`
- 33) Можно ли переписать элементы массива *x* в массив *y* с помощью оператора `y=x`?
- 34) Напишите фрагмент программы ввода массива *B*, описанного следующим образом:
- 35) `typedef int T[6]; T b;`
- 36) Напишите фрагмент программы вывода массива *A*, описанного следующим образом:
- 37) `float a[5];`
- 38) Присвойте нулевые значения всем элементам массива `double a[k];`
- 39) Для чего предназначается компонента `TStringGrid`?

40) Какими свойствами обладает компонента TStringGrid?

3 Вопросы на установление последовательности

Вопрос: Языки программирования в порядке увеличения их уровня 1:

Машинный язык 2: Процедурный язык 3: Объектно-ориентированный язык 4:

Язык ассемблера

1 Ответ1: 1342

2 Ответ2: 1,4,3,2

3 Ответ3: 1234

4 Ответ4: 2134

5 Ответ5: 4321

Вопрос 2: Создание простого класса:

Установите правильную последовательность шагов для создания простого класса в C#.

- a) Определить имя класса
- b) Указать модификатор доступа (например, public)
- c) Открыть фигурные скобки
- d) Закрыть фигурные скобки

Вопрос 3: Создание объекта класса:

Какие шаги нужно выполнить, чтобы создать объект класса?

- a) Написать название класса
- b) Указать имя переменной
- c) Использовать оператор new
- d) Присвоить объект переменной

Цикл for:

Вопрос 4: Расставьте шаги для написания цикла for.

- a) Определить переменную-итератор
- b) Указать условие продолжения цикла
- c) Открыть фигурные скобки
- d) Увеличить переменную-итератор

Вопрос 5: Обработка исключений:

Установите правильную последовательность действий для обработки исключений с помощью try-catch.

- a) Записать код, который может вызвать исключение
- b) Открыть блок try
- c) Написать блок catch
- d) Обработать исключение в блоке catch

Вопрос 6: Использование using:

Расставьте действия, необходимые для использования блока `using` для автоматического освобождения ресурсов.

- a) Создать объект
- b) Открыть блок `using`
- c) Записать код, использующий объект
- d) Закрыть блок `using`

Вопрос 7: Создание свойства:

Каковы шаги для создания свойства в классе?

- a) Указать тип данных
- b) Написать имя свойства
- c) Открыть фигурные скобки
- d) Определить методы `get` и `set`

Вопрос 8: Создание интерфейса:

Установите правильный порядок для создания интерфейса.

- a) Написать ключевое слово `interface`
- b) Указать имя интерфейса
- c) Открыть фигурные скобки
- d) Определить члены интерфейса внутри фигурных скобок

Вопрос 9: Определение и реализация метода:

Какие шаги необходимы для определения и реализации метода в классе?

- a) Указать тип возвращаемого значения
- b) Написать имя метода
- c) Открыть фигурные скобки
- d) Написать код метода

Вопрос 10: Подключение пространства имен:

Каковы шаги для подключения пространства имен в C#?

- a) Написать ключевое слово `using`
- b) Указать имя пространства имен
- c) Завершить строку с помощью точки с запятой

Вопрос 11: Работа с массивами:

Установите последовательность действий для работы с массивами в C#.

- a) Определить тип данных массива
- b) Создать массив с помощью оператора `new`

- c) Записать данные в массив
- d) Использовать массив в коде

Вопрос 12: Создание нового проекта в Visual Studio:

- a) Выберите тип проекта.
- b) Откройте Visual Studio.
- c) Назовите проект.
- d) Нажмите "Создать".

Вопрос 13: Объявление и инициализация переменной:

- a) Укажите тип переменной.
- b) Дайте переменной имя.
- c) Присвойте значение.
- d) Завершите объявление.

Вопрос 14: Создание метода в классе:

- a) Укажите модификатор доступа.
- b) Запишите тип возвращаемого значения.
- c) Укажите имя метода.
- d) Определите параметры метода.
- e) Реализуйте тело метода.

Вопрос 15: Использование foreach для перебора коллекции:

- a) Определите коллекцию.
- b) Начните цикл foreach.
- c) Укажите переменную для текущего элемента.
- d) Поместите код для работы с элементами внутри цикла.
- e) Завершите цикл.

Вопрос 16: Обработка исключений с помощью try-catch:

- a) Напишите код, который может вызвать исключение.
- b) Начните блок try.
- c) Закончите блок try.
- d) Напишите блок catch для обработки исключения.
- e) Завершите блок catch.

Вопрос 17: Создание и использование массива:

- a) Определите размер массива.
- b) Объявите массив с требуемым типом.

- c) Инициализируйте элементы массива.
- d) Используйте массив в вашем коде.

Вопрос 18: Определение и использование интерфейса:

- a) Определите интерфейс.
- b) Укажите его методы и свойства.
- c) Реализуйте интерфейс в классе.
- d) Создайте объект класса.

Вопрос 19: Создание циклического оператора for:

- a) Определите начальное значение счетчика.
- b) Установите условие выхода из цикла.
- c) Установите шаг изменения счетчика.
- d) Напишите тело цикла.

Вопрос 20: Работа с делегатами:

- a) Объявите делегат.
- b) Создайте метод, который соответствует делегату.
- c) Создайте экземпляр делегата.
- d) Вызовите делегат.

Вопрос 21: Создание класса и его объектов:

- a) Определите класс.
- b) Напишите конструктор.
- c) Создайте методы класса.
- d) Создайте экземпляр класса.
- e) Используйте методы экземпляра.

4 Вопросы на установление соответствия

4.1. Установите соответствие программных средств и их назначения:

- 1 – Word
- 2 – Excel
- 3 – PowerPoint
- 4 – Access

А) Работа с базами данных Б)

Работа с текстом

В) Работа с презентациями

Г) Работа с электронными таблицами

4.2 Вопрос: Типы данных

A. int

B. bool

C. string

D. float

Соответствие:

Целое число

Логический тип (истина/ложь)

Последовательность символов

Число с плавающей запятой

1.3. Вопрос: Какой из этих модификаторов/ключевых слов соответствует для объявления неизменяемого поля класса?

Требуемые ключевые слова:

a) readonly

b) const

c) static

1.4. Вопрос: Какой из этих типов данных соответствует ссылочному типу?

a) int

b) string

c) bool

d) float

1.5. Вопрос: Какой из этих циклов соответствует выполнению тела хотя бы один раз?

a) for

b) while

c) do-while

d) foreach

1.6. Вопрос: Какой из этих элементов соответствует для выполнения кода после блока try, независимо от наличия исключений?

Обработка исключений:

- a) try
- b) catch
- c) finally
- d) throw

1.7. Вопрос: Какой из этих модификаторов соответствует доступу к членам класса только внутри той же сборки?

- a) public
- b) private
- c) protected
- d) internal

1.8 Вопрос: Какой из этих типов соответствует для определения контракта, который должны реализовать классы?

Интерфейсы:

- a) interface
- b) abstract class
- c) delegate
- d) enum

4.9. Вопрос: Какой из этих элементов соответствует для создания экземпляра класса?

- a) Конструктор
- b) Деструктор
- c) Метод
- d) Поле

4.10. Вопрос: Как называется механизм, который позволяет контролировать доступ к полям класса и обеспечивать их защиту?

Свойства:

- a) Getter
- b) Setter
- c) Indexers
- d) Events

4.11. Вопрос: Какой из этих делегатов можно использовать для представления метод, который возвращает значение?

- a) Func
- b) Action
- c) Predicate
- d) LINQ

4.12. Вопрос: Какой из этих типов данных обеспечивает хранение пар "ключ-значение"?

Коллекции:

- a) List<T>
- b) ArrayList

c) Dictionary<TKey, TValue>

d) HashSet<T>

4.13 Вопрос: Типы данных

A. int

B. bool

C. string

D. float

Соответствие:

Целое число

Логический тип (истина/ложь)

Последовательность символов

Число с плавающей запятой

4.14 Вопрос: Модификаторы доступа

A. public

B. private

C. protected

D. internal

Соответствие:

Доступен только в пределах класса

Доступен только в пределах сборки

Доступен в производных классах

Доступен из любого другого класса

4.15 Вопрос: Циклы

A. for

B. while

C. do while

D. foreach

Соответствие:

Цикл, который выполняется как минимум один раз

Цикл, который перебирает элементы коллекции

Цикл с заранее определенным числом итераций

Цикл, который выполняется до тех пор, пока условие истинно

4.16 Вопрос: Исключения

A. try

B. catch

C. finally

D. throw

Соответствие:

Блок кода для обработки ошибок

Используется для генерации исключений

Блок кода, который выполняется после попытки

Блок кода, в котором может произойти исключение

4.17 Вопрос: Типы методов

A. void

B. static

C. abstract

D. virtual

Соответствие:

Метод без возвращаемого значения

Метод, который можно переопределить в производном классе

Метод, принадлежащий классу, а не объекту

Метод, который не имеет реализации в базовом классе

4.18 Вопрос: Коллекции

A. ArrayList

B. List<T>

C. Dictionary<TKey, TValue>

D. HashSet<T>

Соответствие:

Неупорядоченная коллекция уникальных элементов

Обобщенная коллекция с ключами и значениями

Коллекция с фиксированным размером

Обобщенная коллекция с динамическим размером

4.19 Вопрос: Операторы

A. ==

B. !=

C. &&

D. ||

Соответствие:

Логический «И»

Логический «ИЛИ»

Оператор равенства

Оператор неравенства

4.20 Вопрос: Классы и объекты

A. class

B. object

C. interface

D. struct

Соответствие:

Пользовательский тип данных с определением методов и свойств

Основной экземпляр класса

Контракт, который реализует класс

Значимый тип, который может содержать методы и поля

4.21 Вопрос: Свойства

A. get

B. set

C. private set

D. public get

Соответствие:

Метод для получения значения свойства

Метод для установки значения свойства

Свойство, доступное только для изменения внутри класса

Свойство, доступное только для чтения

4.22 Вопрос: LINQ

A. Select

B. Where

C. OrderBy

D. GroupBy

Соответствие:

Метод для фильтрации коллекции

Метод для упорядочивания коллекции

Метод для группировки элементов

Метод для проекции данных в новую форму

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) и максимального балла за решение компетентностноориентированной 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 5-балльной шкале следующим образом: 85-100 баллов – отлично, 70-84 балла – хорошо, 50-69 баллов – 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.

2.3 КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ЗАДАЧИ

1 СЕМЕТР

Компетентностно-ориентированная задача №1

Дана вещественная матрица размером 4 строки, 5 столбцов. Переставляя ее строки и столбцы, добейтесь того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.

Компетентностно-ориентированная задача №2

Среди строк целочисленной квадратной матрицы порядка 5 найдите строку с минимальной суммой элементов.

Компетентностно-ориентированная задача №3

Написать программу на нахождение максимального элемента в одномерном массиве.

Компетентностно-ориентированная задача №4

Написать программу по поиску в символьной строке нужного символа.

Компетентностно-ориентированная задача №5

Реализовать математические вычисления при помощи программы на C# по заданию преподавателя.

Компетентностно-ориентированная задача №9

Среди столбцов целочисленной квадратной матрицы порядка 5 найдите столбец с максимальной суммой элементов.

Компетентностно-ориентированная задача №10

В данной вещественной квадратной матрице порядка 5 найдите наименьший элемент. Получите квадратную матрицу порядка 4 путем выбрасывания из исходной матрицы строки и столбца, на пересечении которых расположен элемент с найденным значением.

Компетентностно-ориентированная задача №11

В данной вещественной квадратной матрице порядка 5 найдите среднее геометрическое положительных элементов столбца, в котором расположен элемент с наименьшим значением. Предполагается, что такой элемент единственный.

Компетентностно-ориентированная задача №12

Дана целочисленная матрица размером n строк, m столбцов. Для каждого столбца найдите сумму нечетных элементов и упорядочьте столбцы матрицы по возрастанию сумм.

Компетентностно-ориентированная задача №13

Опишите динамические массивы:

```
float a[5][8], b[7][5];
```

Компетентностно-ориентированная задача №14

Присвойте нулевые значения всем элементам массива: `float a[10][5];`

Компетентностно-ориентированная задача №15

Написать программу по поиску в символьной строке нужного символа.

Компетентностно-ориентированная задача №16

Реализовать объектно-ориентированную программу для формирования расстановки мин на поле для игры «Сапер». Реализовать механизм наследования классов.

Компетентностно-ориентированная задача №17

Разработать объектно-ориентированную программу для конвертации чисел в двоичной системе счисления в шестнадцатиричную и обратно. Преобразование систем счисления реализовать самостоятельно. Реализовать механизм наследования классов.

Компетентностно-ориентированная задача №18

Разработать объектно-ориентированную программу для конвертации чисел в двоичной системе счисления в десятичную и обратно. Преобразование систем счисления реализовать самостоятельно. Реализовать механизм наследования классов.

Компетентностно-ориентированная задача №19

Разработать объектно-ориентированную программу для представления и вывода на экран верхних и нижних индексов символа. Реализовать механизм наследования классов

Компетентностно-ориентированная задача №20

Разработать объектно-ориентированную программу для базовой расстановки фигур на шахматной доске. Для каждой фигуры запрограммировать правила выполнения ходов. Реализовать механизм наследования классов.

2 СЕМЕТР

Компетентностно-ориентированная задача №1

Нарисовать изображение по заданию преподавателя на языке C#.

Компетентностно-ориентированная задача №2

Выполнить операции с файлом данных по заданию преподавателя.

Компетентностно-ориентированная задача №3

В окно светло-голубого цвета выведите желтый круг диаметром 400 пикселей, с «выеденной» с правой стороны частью (как у надкусанного яблока).

Выведите поверх изображения свою фамилию шрифтом Times размером 80 пикселей, подчеркнутым, жирным, курсивом, красного цвета. Надпись расположите под углом 75° (снизу вверх).

Компетентностно-ориентированная задача №4

В окно голубого цвета выведите изображение Солнца в виде желтого круга, эллиптической орбиты Земли вокруг него и самой Земли в виде коричневого кружка в верхней точке этой орбиты. Выведите поверх изображения горизонтально свою фамилию шрифтом Courier размером 20 пикселей, подчеркнутым, белого цвета. Надпись расположите горизонтально.

Компетентностно-ориентированная задача №5

В окно желтого цвета выведите неправильный выпуклый пятиугольник темно-синего цвета. Такая фигура рисуется с помощью функции Polygon() класса

TCanvas. Выведите поверх изображения свою фамилию шрифтом Arial размером 10 пикселей, красного цвета. Надпись расположите горизонтально «вверх ногами».

Компетентностно-ориентированная задача №6

Реализовать математические вычисления по и нарисовать график функции на языке C#.

Компетентностно-ориентированная задача №7

Структуры. Имеются данные о сотрудниках фирмы (фамилия, зарплата, пол). Найти фамилию мужчины, имеющего самую высокую зарплату. Если таких сотрудников несколько, то вывести всех, упорядочив список фамилий по возрастанию.

Компетентностно-ориентированная задача №8

Структуры. Имеются данные о сотрудниках фирмы (фамилия, возраст, отношение к воинской службе). Найти фамилию сотрудника, младшего по возрасту и невоеннообязанного. Если таких сотрудников несколько, то вывести всех, упорядочив список фамилий по возрастанию.

Компетентностно-ориентированная задача №9

Структуры. Известна информация о багаже пассажиров (фамилия, количество вещей и общий вес багажа). Найти количество пассажиров и их фамилии, вес багажа которых превышает среднее значение багажа всех пассажиров.

Компетентностно-ориентированная задача №10

Структуры. Известны данные о трех оценках учеников класса (фамилия, оценки). Вывести фамилии учеников, имеющие сумму оценок, занимающую второе место в классе.

Компетентностно-ориентированная задача №11

Структуры. Имеются данные о днях месяца (температура воздуха, количество осадков). Определить количество осадков, выпавших в виде дождя (считать, что идет дождь, если температура больше 0).

Компетентностно-ориентированная задача №12

Написать программу, которая выполняет с файлом следующие действия:

- создание файла;
- добавление записи в файл;
- просмотр содержимого файла;
- удаление записи из файла;
- поиск записи в файле.

Имеются результаты медицинского обследования группы детей, в которых указаны: фамилия, адрес, возраст, вес, рост и наличие прививок по возрасту.

Компетентностно-ориентированная задача №13

Написать программу, которая выполняет с файлом следующие действия:

- создание файла;
- добавление записи в файл;
- просмотр содержимого файла;

- удаление записи из файла;
- поиск записи в файле.

В библиотеке хранятся сведения о книгах: название книги, автор, год выпуска, стоимость, издательство, как часто книга пользовалась спросом у читателей.

Компетентностно-ориентированная задача №14

Написать программу, которая выполняет с файлом следующие действия:

- создание файла;
- добавление записи в файл;
- просмотр содержимого файла;
- удаление записи из файла;
- поиск записи в файле.

В кинотеатрах города демонстрируются фильмы. Известны названия кинотеатров, названия фильмов, киностудия, режиссёр, дата первого показа и дата последнего показа.

Компетентностно-ориентированная задача №15

Написать программу, которая выполняет с файлом следующие действия:

- создание файла;
- добавление записи в файл;
- просмотр содержимого файла;
- удаление записи из файла;
- поиск записи в файле.

В администрации хранятся сведения о фирмах: название, дата регистрации, форма собственности, № счёта, сумма доходов за предыдущий год, количество сотрудников.

Компетентностно-ориентированная задача №16

Даны два текстовых файла. Добавить в начало первого файла содержимое второго файла

Компетентностно-ориентированная задача №17

Дано целое число k и текстовый файл. Вставить пустую строку перед строкой файла с номером k . Если строки с таким номером нет, то поставить пустую строку в начало файла.

Компетентностно-ориентированная задача №18

Дано целое число k и текстовый файл. Удалить из файла строку с номером k . Если строки с таким номером нет, то оставить файл без изменений.

Компетентностно-ориентированная задача №19

Дан текстовый файл. Заменить в нем все подряд идущие пробелы на один пробел.

Компетентностно-ориентированная задача №20

Дано целое число k ($0 < k \leq 10$) и текстовый файл, содержащий более k строк. Создать новый текстовый файл, содержащий k последних строк исходного файла.

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 компетентностноориентированной задачи – 6 баллов.

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

Общий балл по промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по 5-балльной шкале следующим образом: 85-100 баллов – отлично, 70-84 балла – хорошо, 50-69 баллов – удовлетворительно, 49 и менее – неудовлетворительно.

Критерии оценивания решения компетентностно-ориентированной задачи

6-5 баллов выставляется обучающемуся, если решение задачи демонстрирует глубокое понимание обучающимся предложенной проблемы и разностороннее ее рассмотрение; свободно конструируемая работа представляет собой логичное, ясное и при этом краткое, точное описание хода решения задачи (последовательности (или выполнения) необходимых трудовых действий) и формулировку доказанного, правильного вывода (ответа); при этом обучающимся предложено несколько вариантов решения или оригинальное, нестандартное решение (или наиболее эффективное, или наиболее рациональное, или оптимальное, или единственно правильное решение); задача решена в установленное преподавателем время или с опережением времени.

4-3 балла выставляется обучающемуся, если решение задачи демонстрирует понимание обучающимся предложенной проблемы; задача решена типовым способом в установленное преподавателем время; имеют место общие фразы и (или) несущественные недочеты в описании хода решения и (или) вывода (ответа).

2-1 балла выставляется обучающемуся, если решение задачи демонстрирует поверхностное понимание обучающимся предложенной проблемы; осуществлена попытка шаблонного решения задачи, но при ее решении допущены ошибки и (или) превышено установленное преподавателем время.

0 баллов выставляется обучающемуся, если решение задачи демонстрирует непонимание обучающимся предложенной проблемы, и (или) значительное место занимают общие фразы и голословные рассуждения, и (или) задача не решена.