

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Дмитриев Николай Николаевич
Должность: Ректор
Дата подписания: 14.02.2025 08:09:17
Уникальный программный ключ:
f7c6227919e4cdbfb4d7b682991f8553b37cafbf

Министерство сельского хозяйства Российской Федерации
Иркутский государственный аграрный университет
имени А.А. Ежевского

Колледж автомобильного транспорта и агротехнологий

УТВЕРЖДАЮ:
Директор



Н.Н. Бельков
«31» марта 2023 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.02 Осуществление интеграции программных модулей**

Специальность: 09.02.07 Информационные системы и программирование
(программа подготовки специалистов среднего звена)

Форма обучения: очная
II, III курсы; 4, 6 семестры

Молодежный 2023

1. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

Фонд оценочных средств для промежуточной аттестации по профессиональному модулю ПМ.02 Осуществление интеграции программных модулей, включает:

- перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения (промежуточной аттестации) по дисциплине, характеризующих этапы формирования компетенций и (или) для итогового контроля сформированности компетенций.

2. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Рабочая программа профессионального модуля определяет перечень планируемых результатов обучения модулю, соотнесенных с планируемыми результатами освоения образовательной программы.

Код	Наименование компетенции (планируемые результаты освоения ОП)	Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенции
	Общие компетенции	В области знания и понимания (А)
Вид деятельности: программных модулей	Осуществление интеграции	Уметь: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы;
ОК 01	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.	Знать: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте;

		алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.	<p>Уметь: определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска</p> <p>Знать: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации</p>
	Профессиональные компетенции	В области интеллектуальных навыков (В)

<p>ПК 2.1.</p>	<p>Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.</p>	<p>Уметь: Анализировать проектную и техническую документацию. Использовать специализированные графические средства построения и анализа архитектуры программных продуктов. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Определять источники и приемники данных. Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace). Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выявлять ошибки в системных компонентах на основе спецификаций.</p> <p>Знать: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и варианты интеграционных решений. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы отладочных классов. Стандарты качества программной документации.</p>
----------------	---	---

		<p>Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Графические средства проектирования архитектуры программных продуктов. Методы организации работы в команде разработчиков. Практический опыт: Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>
ПК 2.2.	Выполнять интеграцию модулей в программное обеспечение.	<p>Уметь: Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Использовать различные транспортные протоколы и стандарты форматирования сообщений. Выполнять тестирование интеграции. Организовывать постобработку данных. Создавать классы-исключения на основе базовых классов. Выполнять ручное и</p>

		<p>автоматизированное тестирование программного модуля.</p> <p>Выявлять ошибки в системных компонентах на основе спецификаций.</p> <p>Использовать приемы работы в системах контроля версий.</p> <p>Знать:</p> <p>Модели процесса разработки программного обеспечения.</p> <p>Основные принципы процесса разработки программного обеспечения.</p> <p>Основные подходы к интегрированию программных модулей.</p> <p>Основы верификации программного обеспечения.</p> <p>Современные технологии и инструменты интеграции.</p> <p>Основные протоколы доступа к данным.</p> <p>Методы и способы идентификации сбоев и ошибок при интеграции приложений.</p> <p>Основные методы отладки.</p> <p>Методы и схемы обработки исключительных ситуаций.</p> <p>Основные методы и виды тестирования программных продуктов.</p> <p>Стандарты качества программной документации.</p> <p>Основы организации инспектирования и верификации.</p> <p>Приемы работы с инструментальными средствами тестирования и отладки.</p> <p>Методы организации работы в команде разработчиков.</p> <p>Практический опыт:</p> <p>Интегрировать модули в программное обеспечение.</p> <p>Отлаживать программные модули.</p> <p>Инспектировать разработанные программные модули на предмет</p>
--	--	--

		соответствия стандартам кодирования.
ПК 2.3.	Выполнять отладку программного модуля с использованием специализированных программных средств.	<p>Уметь: Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Использовать инструментальные средства отладки программных продуктов. Определять источники и приемники данных. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Выполнять отладку, используя методы и инструменты условной компиляции. Выявлять ошибки в системных компонентах на основе спецификаций.</p> <p>Знать: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Основные методы отладки. Методы и схемы обработки исключительных ситуаций. Приемы работы с инструментальными средствами тестирования и</p>

		<p>отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков. Практический опыт: Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>
ПК 2.4.	<p>Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.</p>	<p>Уметь: Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций. Знать: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и</p>

		<p>аттестации программного обеспечения.</p> <p>Методы и способы идентификации сбоев и ошибок при интеграции приложений.</p> <p>Методы и схемы обработки исключительных ситуаций.</p> <p>Основные методы и виды тестирования программных продуктов.</p> <p>Приемы работы с инструментальными средствами тестирования и отладки.</p> <p>Стандарты качества программной документации.</p> <p>Основы организации инспектирования и верификации.</p> <p>Встроенные и основные специализированные инструменты анализа качества программных продуктов.</p> <p>Методы организации работы в команде разработчиков.</p> <p>Практический опыт:</p> <p>Разрабатывать тестовые наборы (пакеты) для программного модуля.</p> <p>Разрабатывать тестовые сценарии программного средства.</p> <p>Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>
ПК 2.5.	<p>Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.</p>	<p>Уметь:</p> <p>Использовать выбранную систему контроля версий.</p> <p>Использовать методы для получения кода с заданной функциональностью и степенью качества.</p> <p>Анализировать проектную и техническую документацию.</p> <p>Организовывать постобработку данных.</p> <p>Приемы работы в системах контроля версий.</p> <p>Выявлять ошибки в системных</p>

		<p>компонентах на основе спецификаций.</p> <p>Знать: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.</p> <p>Практический опыт: Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>
--	--	--

3.ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ

При проведении промежуточной аттестации в колледже используются традиционные формы аттестации:

Элемент модуля	Форма промежуточной аттестации	Шкала оценивания
МДК 02.01 Технология разработки программного обеспечения	Экзамен Контрольные работы	"отлично", "хорошо", "удовлетворительно", "неудовлетворительно"
МДК 02.02 Инструментальные средства разработки программного обеспечения	Экзамен Курсовая работа	"отлично", "хорошо", "удовлетворительно", "неудовлетворительно"
МДК 02.03 Математическое моделирование	Контрольные работы	

4. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ (ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ И (ИЛИ) ДЛЯ ИТОГОВОГО КОНТРОЛЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИИ

4.1. Примерный перечень тестов к экзамену МДК 02.01 Технология разработки программного обеспечения (семестр 4)

1. Какие программы можно отнести к системному программному обеспечению:
Варианты ответа:
 1. операционные системы;
 2. прикладные программы;
 3. игровые программы.
2. Какие программы можно отнести к системному ПО:
Варианты ответа:
 - 1) драйверы;
 - 2) текстовые редакторы;
 - 3) электронные таблицы;
 - 4) графические редакторы.
3. Специфические особенности ПО как продукта:
 - 1) продажа по ценам ниже себестоимости (лицензирование);
 - 2) низкие материальные затраты при создании программ;
 - 3) возможность создание программ небольшие коллективом или даже одним человеком;
 - 4) разнообразие решаемых задач с помощью программных средств.
4. Какие программы можно отнести к системному ПО:
Варианты ответа:
 - 1) программа расчета заработной платы;
 - 2) электронные таблицы;
 - 3) СУБД (системы управления базами данных).
5. Какие программы нельзя отнести к системному ПО:
Варианты ответа:
 - 1) игровые программы;
 - 2) компиляторы языков программирования;
 - 3) операционные системы;
 - 4) системы управления базами данных.
6. Какие программы можно отнести к прикладному программному обеспечению:
Варианты ответа:
 - 1) электронные таблицы;
 - 2) таблицы решений;
 - 3) СУБД (системы управления базами данных).
7. Какие программы можно отнести к прикладному ПО:
Варианты ответа:
 - 1) программа расчета заработной платы;
 - 2) диспетчер программ;
 - 3) программа «Проводник» (Explorer).

8. Какие программы нельзя отнести к прикладному ПО:

Варианты ответа:

- 1) компиляторы и (или) интерпретаторы;
- 2) текстовые и (или) графические редакторы;
- 3) электронные таблицы.

9. Можно ли отнести операционную систему к программному обеспечению:

Варианты ответа:

- 1) да;
- 2) нет.

10. Можно ли отнести операционную систему к прикладному программному обеспечению:

Варианты ответа:

- 1) да;
- 2) нет.

11. Специфические особенности ПО как продукта:

Варианты ответа:

- 1) низкие затраты при дублировании;
- 2) универсальность;
- 3) простота эксплуатации;
- 4) наличие поддержки (сопровождения) со стороны разработчика.

12. Какие программы можно отнести к системному ПО:

Варианты ответа:

- 1) утилиты;
- 2) экономические программы;
- 3) статистические программы;
- 4) мультимедийные программы.

13. Этап, занимающий наибольшее время, в жизненном цикле программы:

Варианты ответа:

- 1) сопровождение;
- 2) проектирование;
- 3) тестирование;
- 4) программирование;
- 5) формулировка требований.

14. Этап, занимающий наибольшее время, при разработке программы:

Варианты ответа:

- 1) тестирование;
- 2) сопровождение;
- 3) проектирование;
- 4) программирование;
- 5) формулировка требований.

15. Первый этап в жизненном цикле программы:

Варианты ответа:

- 1) формулирование требований;
- 2) анализ требований;
- 3) проектирование;
- 4) автономное тестирование;
- 5) комплексное тестирование.

16. Один из необязательных этапов жизненного цикла программы:

Варианты ответа:

- 1) оптимизация;
- 2) проектирование;
- 3) тестирование;
- 4) программирование;
- 5) анализ требований.

17. Самый большой этап в жизненном цикле программы:

Варианты ответа:

- 1) эксплуатация;
- 2) изучение предметной области;
- 3) программирование;
- 4) тестирование;
- 5) корректировка ошибок.

18. Какой этап выполняется раньше:

Варианты ответа:

- 1) отладка;
- 2) тестирование.

19. Какой этап выполняется раньше:

Варианты ответа:

- 1) отладка;
- 2) оптимизация;
- 3) программирование;
- 4) тестирование.

20. Что выполняется раньше:

Варианты ответа:

- 1) компиляция;
- 2) отладка;
- 3) компоновка;
- 4) тестирование.

21. Что выполняется раньше:

Варианты ответа:

- 1) проектирование;
- 2) программирование;
- 3) отладка;
- 4) тестирование.

22. В стадии разработки программы не входит:

Варианты ответа:

- 1) автоматизация программирования;
- 2) постановка задачи;
- 3) составление спецификаций;
- 4) эскизный проект;
- 5) тестирование.

23. Самый важный критерий качества программы:

Варианты ответа:

- 1) работоспособность;
- 2) надежность;
- 3) эффективность;
- 4) быстроедействие;
- 5) простота эксплуатации.

24. Способы оценки качества:

Варианты ответа:

- 1) сравнение с аналогами;
- 2) наличие документации;
- 3) оптимизация программы;
- 4) структурирование алгоритма.

25. Существует ли связь между эффективностью и оптимизацией программы:

Варианты ответа:

- 1) да;
- 2) нет.

26. Наиболее важный критерий качества:

Варианты ответа:

- 1) надежность;
- 2) быстроедействие;
- 3) удобство в эксплуатации;
- 4) удобный интерфейс;
- 5) эффективность.

27. Способы оценки надежности:

Варианты ответа:

- 1) тестирование;
- 2) сравнение с аналогами;
- 3) трассировка;
- 4) оптимизация.

28. Повышает ли качество программ оптимизация:

Варианты ответа:

- 1) да;
- 2) нет.

29. Существует ли связь между надежностью и быстрымдействием:

Варианты ответа:

- 1) нет;
- 2) да.

30. В каких единицах можно измерить надежность:

Варианты ответа:

- 1) отказов/час;
- 2) км/час;
- 3) Кбайт/сек;
- 4) операций/сек.

31. В каких единицах можно измерить быстроедействие:

Варианты ответа:

- 1) отказов/час;
- 2) км/час;
- 3) Кбайт/сек;
- 4) операций/сек.

32. Что относится к этапу программирования:

Варианты ответа:

- 1) написание кода программы;
- 2) В) разработка интерфейса;
- 3) С) работоспособность;
- 4) анализ требований.

33. Последовательность этапов программирования:

Варианты ответа:

- 1) компилирование, компоновка, отладка;
- 2) В) компоновка, отладка, компилирование;

3) отладка, компилирование, компоновка;

4) компилирование, отладка, компоновка.

34) Инструментальные средства программирования:

Варианты ответа:

1) компиляторы, интерпретаторы;

2) СУБД (системы управления базами данных);

3) BIOS (базовая система ввода-вывода);

4) ОС (операционные системы).

35. На языке программирования составляется:

Варианты ответа:

1) исходный код;

2) исполняемый код;

3) объектный код;

4) алгоритм.

36. Правила, которым должна следовать программа это:

Варианты ответа:

1) алгоритм;

2) структура;

3) спецификация;

4) состав информации.

37. Можно ли внутри цикла поместить еще один цикл:

Варианты ответа:

1) да;

2) нет.

38. Можно ли внутри условного оператора поместить еще одно условие:

Варианты ответа:

1) да;

2) нет.

39. Можно ли одно большое (длинное) выражение разбить на несколько выражений:

Варианты ответа:

1) да;

2) нет.

40. Если имеется стандартная функция, нужно ли писать собственную:

Варианты ответа:

1) нет;

2) да.

41. Доступ, при котором записи файла читаются в физической последовательности, называется:

Варианты ответа:

1) последовательным;

2) прямым;

3) простым;

4) основным.

42. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:

Варианты ответа:

1) прямым;

2) последовательным;

3) простым;

4) основным.

43. Методы программирования (укажите НЕ верный ответ):

Варианты ответа:

- 1) логическое;
- 2) структурное;
- 3) модульное.

44. Что выполняется раньше:

Варианты ответа:

- 1) разработка алгоритма;
- 2) выбор языка программирования;
- 3) написание исходного кода;
- 4) компиляция.

45. Можно ли переменным присваивать произвольные идентификаторы:

Варианты ответа:

- 1) да;
- 2) нет.

46. Найдите НЕ правильное условие для создания имен:

Варианты ответа:

- 1) имена могут содержать пробелы;
- 2) длинное имя можно сократить;
- 3) из имени лучше выбрасывать гласные;
- 4) можно использовать большие буквы.

47. Какие символы не допускаются в именах переменных:

Варианты ответа:

- 1) пробелы;
- 2) цифры;
- 3) подчеркивание

48. Можно ли использовать имена, которые уже были использованы в другой программе (модуле):

Варианты ответа:

- 1) да;
- 2) нет.

49. Можно ли ставить знак подчеркивания в начале имени:

Варианты ответа:

- 1) да, но не рекомендуется;
- 2) да, без ограничений;
- 3) нет.

50. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:

Варианты ответа:

- 1) прямым указанием;
- 2) венгерской нотацией;
- 3) структурным программированием;
- 4) поляризацией.

51. Можно ли писать комментарии в отдельной строке:

Варианты ответа:

- 1) да;
- 2) нет.

52. Транслируются ли комментарии:

Варианты ответа:

- 1) да;
- 2) нет.

53. Наличие комментариев позволяет:

Варианты ответа:

- 1) быстрее найти ошибки в программе;
- 2) быстрее писать программы;
- 3) быстрее выполнять программы.

54. Наличие комментариев позволяет:

Варианты ответа:

- 1) легче разобраться в программе;
- 2) применять сложные структуры;
- 3) увеличить быстродействие.

55. Наличие комментариев позволяет:

Варианты ответа:

- 1) улучшить читабельность программы;
- 2) улучшить эксплуатацию программы;
- 3) повысить надежность программы.

56. Что определяет выбор языка программирования:

Варианты ответа:

- 1) область приложения;
- 2) знание языка;
- 3) наличие дополнительных библиотек.

57. Возможно ли комбинирование языков программирования в рамках одной задачи:

Варианты ответа:

- 1) да;
- 2) нет.

58. Допустимо ли комбинирование языков программирования в рамках одной задачи :

Варианты ответа:

- 1) да;
- 2) нет.

59. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:

Варианты ответа:

- 1) для экономических задач;
- 2) для системных задач;
- 3) для инженерных задач.

60. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:

Варианты ответа:

- 1) для инженерных задач;
- 2) для системных задач;
- 3) для экономических задач.

61. На каком этапе производится выбор языка программирования:

Варианты ответа:

- 1) проектирование;
- 2) программирование;
- 3) отладка;
- 4) тестирование.

62. Можно ли использовать комбинацию языков программирования в рамках одного проекта:

Варианты ответа:

- 1) да;
- 2) нет.

63. Для решения экономических задач характерно применение:

Варианты ответа:

- 1) СУБД (систем управления базами данных);
- 2) языков высокого уровня;
- 3) языков низкого уровня;
- 4) применение сложных математических расчетов.

64. Для решения инженерных задач характерно применение:

Варианты ответа:

- 1) САПР (систем автоматизированного проектирования);
- 2) СУБД (систем управления базами данных);
- 3) ОС (операционных систем).

65. Причины синтаксических ошибок:

Варианты ответа:

- 1) плохое знание языка программирования;
- 2) ошибки в исходных данных;
- 3) ошибки, допущенные на более ранних этапах;
- 4) неправильное применение процедуры тестирования.

66. Когда можно обнаружить синтаксические ошибки:

Варианты ответа:

- 1) при компиляции;
- 2) при отладке;
- 3) при тестировании;
- 4) на этапе проектирования;
- 5) при эксплуатации.

67. Ошибки компоновки заключаются в том, что:

Варианты ответа:

- 1) указано внешнее имя, но не объявлено;
- 2) неправильно использовано зарезервированное слово;
- 3) составлено неверное выражение;
- 4) указан неверный тип переменной.

68. Могут ли проявиться ошибки при изменении условий эксплуатации:

Варианты ответа:

- 1) да;
- 2) нет.

69. Могут ли проявиться ошибки при изменении в предметной области:

Варианты ответа:

- 1) да;
- 2) нет.

70. Возможно ли программирование с защитой от ошибок:

Варианты ответа:

- 1) да;
- 2) нет.

71. Есть ли недостатки программирования с защитой от ошибок:

Варианты ответа:

- 1) да;
- 2) нет.

72. Защитное программирование это:

Варианты ответа:

- 1) встраивание в программу отладочных средств;
- 2) создание задач защищенных от копирования;
- 3) разделение доступа в программе;
- 4) использование паролей;
- 5) оформление авторских прав на программу.

73. Вид ошибки с неправильным написанием служебных слов (операторов):

Варианты ответа:

- 1) синтаксическая;
- 2) семантическая;
- 3) логическая;
- 4) символьная.

74. Вид ошибки с неправильным использованием служебных слов (операторов):

Варианты ответа:

- 1) семантическая;
- 2) синтаксическая;
- 3) логическая;
- 4) символьная.

75. Ошибки при написании программы бывают:

Варианты ответа:

- 1) синтаксические;
- 2) орфографические;
- 3) лексические;
- 4) фонетические;
- 5) морфологические.

76. Процедура поиска ошибки, когда известно, что она есть это:

Варианты ответа:

- 1) отладка;
- 2) тестирование;
- 3) компоновка;
- 4) транзакция;
- 5) трансляция.

77. Программа для просмотра значений переменных при выполнении программы:

Варианты ответа:

- 1) отладчик;
- 2) компилятор;
- 3) интерпретатор;
- 4) трассировка;
- 5) тестирование.

78. Отладка – это:

Варианты ответа:

- 1) процедура поиска ошибок, когда известно, что ошибка есть;
- 2) определение списка параметров;
- 3) правило вызова процедур (функций);
- 4) составление блок-схемы алгоритма.

79. Когда программист может проследить последовательность выполнения команд программы:

Варианты ответа:

- 1) при трассировке;
- 2) при тестировании;
- 3) при компиляции;

- 4) при выполнении программы;
- 5) при компоновке.

80. На каком этапе создания программы могут появиться синтаксические ошибки:

Варианты ответа:

- 1) программирование;
- 2) проектирование;
- 3) анализ требований;
- 4) тестирование.

81. Когда приступают к тестированию программы:

Варианты ответа:

- 1) когда программа уже закончена;
- 2) после постановки задачи;
- 3) на этапе программирования;
- 4) на этапе проектирования;
- 5) после составления спецификаций,

82. Тестирование бывает:

Варианты ответа:

- 1) автономное;
- 2) инструментальное;
- 3) визуальное;
- 4) алгоритмическое.

83. Тестирование бывает:

Варианты ответа:

- 1) комплексное;
- 2) инструментальное;
- 3) визуальное;
- 4) алгоритмическое.

84. Существует ли различие между отладкой и тестированием:

Варианты ответа:

- 1) да;
- 2) нет.

85. При комплексном тестировании проверяются:

Варианты ответа:

- 1) согласованность работы отдельных частей программы;
- 2) правильность работы отдельных частей программы;
- 3) быстродействие программы;
- 4) эффективность программы.

86. Чему нужно уделять больше времени, чтобы получить хорошую программу:

Варианты ответа:

- 1) тестированию;
- 2) программированию;
- 3) отладке;
- 4) проектированию.

87. Процесс исполнения программы с целью обнаружения ошибок:

Варианты ответа:

- 1) тестирование;
- 2) кодирование;
- 3) сопровождение;
- 4) проектирование.

88. Автономное тестирование это:

Варианты ответа:

- 1) тестирование отдельных частей программы;

- 2) инструментальное средство отладки;
- 3) составление блок-схем;
- 4) пошаговая проверка выполнения программы.

89. Трассировка это:

Варианты ответа:

- 1) проверка пошагового выполнения программы;
- 2) тестирование исходного кода;
- 3) отладка модуля;
- 4) составление блок-схемы алгоритма.

90. Локализация ошибки:

Варианты ответа:

- 1) определение места возникновения ошибки;
- 2) определение причин ошибки;
- 3) обнаружение причин ошибки;
- 4) исправление ошибки.

91. Назначение тестирования:

Варианты ответа:

- 1) повышение надежности программы;
- 2) обнаружение ошибок;
- 3) повышение эффективности программы;
- 4) улучшение эксплуатационных характеристик;
- 5) приведение программы к структурированному виду.

92. Назначение отладки:

Варианты ответа:

- 1) поиск причин существующих ошибок;
- 2) поиск возможных ошибок;
- 3) составление спецификаций;
- 4) разработка алгоритма.

93. Инструментальные средства отладки (НЕ правильный ответ):

Варианты ответа:

- 1) компиляторы;
- 2) отладчики;
- 3) трассировка.

94. Отладка программ это:

Варианты ответа:

- 1) локализация и исправление ошибок;
- 2) алгоритмизация программирования;
- 3) компиляция и компоновка.

95. Что выполняется раньше, автономная или комплексная отладка:

Варианты ответа:

- 1) автономная;
- 2) комплексная.

96. Что выполняется раньше, отладка или тестирование:

Варианты ответа:

- 1) отладка;
- 2) тестирование.

97. Что такое автоматизация программирования:

Варианты ответа:

- 1) создание исходного кода программными средствами;
- 2) создание исходного кода при помощи компилятора;

3) создание исходного кода без разработки алгоритма.

98. В чем сущность автоматизации программирования:

Варианты ответа:

- 1) создание программы без написания ее текста;
- 2) получение готовой программы без выполнения компоновки;
- 3) в отсутствии компиляции.

99. Возможна ли автоматизация программирования:

Варианты ответа:

- 1) да;
- 2) нет.

100. Создание исполняемого кода программы без написания исходного кода называется:

Варианты ответа:

- 1) составлением спецификаций;
- 2) отладкой;
- 3) проектированием.
- 4) автоматизацией программирования;

101. Одно из преимуществ автоматизации программирования:

Варианты ответа:

- 1) наглядное программирование с визуальным контролем;
- 2) получение стандартной программы;
- 3) создание программы с оптимальным кодом.

102. Один из методов автоматизации программирования:

Варианты ответа:

- 1) структурное программирование;
- 2) модульное программирование;
- 3) визуальное программирование;
- 4) объектно-ориентированное программирование.

103. Влияет ли автоматизация программирования на эффективность программы:

Варианты ответа:

- 1) нет;
- 2) да

104. Автоматизация программирования позволяет:

Варианты ответа:

- 1) повысить надежность программы;
- 2) сократить время разработки программы;
- 3) повысить быстродействие программы.

105. Позволяет ли автоматизация программирования всегда создавать эффективные программы:

Варианты ответа:

- 1) да.
- 2) нет;

106. Позволяет ли автоматизация программирования всегда создавать надежные программы:

Варианты ответа:

- A) нет;
- B) да.

107. Недостаток автоматизации программирования:

Варианты ответа:

- В) низкое быстродействие;
- А) большой размер программы;
- С) сложность программы.

108. Возможны ли ошибки при автоматизации программирования:

Варианты ответа:

- А) да;
- В) нет.

109. Нужно ли выполнять тестирование при автоматизации программирования:

Варианты ответа:

- А) да;
- В) нет.

110. Выполняется ли процедура компиляции при автоматизации программирования:

Варианты ответа:

- А) да;
- В) нет.

111. Что легко поддается автоматизации:

Варианты ответа:

- А) интерфейс;
- В) работа с файлами;
- С) сложные логические задачи;
- Д) алгоритмизация.

112. Относится ли визуальное программирование к средствам автоматизации:

Варианты ответа:

- А) да;
- В) нет.

113. Нахождение наилучшего варианта из множества возможных:

Варианты ответа:

- А) оптимизация;
- В) тестирование;
- С) автоматизация;
- Д) отладка;
- Е) сопровождение.

114. Что такое оптимизация программ:

Варианты ответа:

- А) улучшение работы существующей программы;
- В) создание удобного интерфейса пользователя;
- С) разработка модульной конструкции программы;
- Д) применение методов объектно-ориентированного программирования.

115. Критерии оптимизации:

Варианты ответа:

- А) время выполнения или размер требуемой памяти;
- В) размер программы и ее эффективность;
- С) независимость модулей;
- Д) качество программы, ее надежность.

116. Критерии оптимизации:

Варианты ответа:

- А) эффективность использования ресурсов;

- В) структурирование алгоритма;
- С) структурирование программы.

117. Возможна ли оптимизация программ без участия программиста:

Варианты ответа:

- А) да;
- В) нет.

118. Возможна ли оптимизация циклов:

Варианты ответа:

- А) да;
- В) нет.

119. В чем заключается оптимизация условных выражений:

Варианты ответа:

- А) в изменении порядка следования элементов выражения;
- В) в использовании простых логических выражений;
- С) в использовании сложных логических выражений;
- Д) в использовании операций AND, OR и NOT.

120. Оптимизация циклов заключается в:

Варианты ответа:

- А) уменьшении количества повторений тела цикла;
- В) просмотре задачи с другой стороны;
- С) упрощение задачи за счет включения логических операций.

121. Оптимизация программы это:

Варианты ответа:

- А) модификация;
- В) отладка;
- С) повышение сложности программы;
- Д) уменьшение сложности программы.

122. Критерии оптимизации программы:

Варианты ответа:

- А) быстродействие или размер программы;
- В) быстродействие и размер программы;
- С) надежность или эффективность;
- Д) надежность и эффективность.

123. Результат оптимизации программы:

Варианты ответа:

- А) эффективность;
- В) надежность;
- С) машино-независимость;
- Д) мобильность.

124. Сущность оптимизации циклов:

Варианты ответа:

- А) сокращение количества повторений выполнения тела цикла;
- В) сокращение тела цикла;
- С) представление циклов в виде блок-схем;
- Д) трассировка циклов;
- Е) поиск ошибок в циклах.

125. В чем сущность модульного программирования:

Варианты ответа:

- А) в разбиении программы на отдельные функционально независимые части;
- В) в разбиении программы на отдельные равные части;
- С) в разбиение программы на процедуры и функции;

126. Можно ли сочетать модульное и структурное программирование:

Варианты ответа:

- A) да;
- B) нет.

127. Может ли модуль включать несколько процедур или функций:

Варианты ответа:

- A) да;
- B) нет.

128. Рекомендуемые размеры модулей:

Варианты ответа:

- A) небольшие;
- B) большие;
- C) равные;
- D) фиксированной длины.

129. В чем заключается независимость модуля:

Варианты ответа:

- A) в написании, отладке и тестировании независимо от остальных модулей;
- B) в разработке и написании независимо от других модулей;
- C) в независимости от работы основной программы.

130. При модульном программировании желательно, чтобы модуль имел:

Варианты ответа:

- A) большой размер;
- B) небольшой размер;
- C) фиксированный размер;
- D) любой размер.

131. Модульное программирование это:

Варианты ответа:

- A) разбиение программы на отдельные части;
- B) структурирование;
- C) использование стандартных процедур и функций.

132. Можно ли использовать оператор GO TO в модульных программах:

Варианты ответа:

- A) можно;
- B) нельзя.

133. Разрешается ли использование циклов при модульном программировании:

Варианты ответа:

- A) да;
- B) нет.

134. Разрешается ли использование условных операторов при модульном программировании:

Варианты ответа:

- A) да;
- B) нет.

135. Сократится ли размер программы, если ее написать в виде набора модулей:

Варианты ответа:

- A) нет;
- B) да.

136. Достоинство модульного программирования:

Варианты ответа:

- A) создание программы по частям в произвольном порядке;

- В) не требует компоновки;
- С) всегда дает эффективные программы;
- Д) снижает количество ошибок.

137. Недостаток модульного программирования:

Варианты ответа:

- А) увеличивает трудоемкость программирования;
- В) усложняет процедуру комплексного тестирования;
- С) снижает быстродействие программы;
- Д) не позволяет выполнять оптимизацию программы.

138. Достоинство модульного программирования:

Варианты ответа:

- А) возможность приступить к тестированию до завершения написания всей программы;
- В) не требует комплексного тестирования;
- С) уменьшает размер программы;
- Д) повышает надежность программы.

139. Допустимо ли использование оператора GO TO при структурном программировании:

Варианты ответа:

- А) нет;
- В) да.

140. Можно ли сочетать структурное программирование с модульным:

Варианты ответа:

- А) можно;
- В) нельзя;
- С) только в особых случаях.

141. Любую ли программу можно привести к структурированному виду:

Варианты ответа:

- А) любую;
- В) не все;
- С) нельзя.

142. Можно ли использовать оператор GO TO в структурированных программах:

Варианты ответа:

- А) можно;
- В) нельзя;
- С) только в особых случаях.

143. Возможно, ли преобразовать неструктурированную программу к структурному виду:

Варианты ответа:

- А) да;
- В) нет.

144. Возможно ли программирование без оператора GO TO:

Варианты ответа:

- А) да;
- В) нет.

145. При структурном программировании задача выполняется:

Варианты ответа:

- А) поэтапным разбиением на более легкие задачи;
- В) без участия программиста;
- С) объединением отдельных модулей программы.

146. Разрешается ли использование оператора GO TO при структурном программировании:

Варианты ответа:

- A) нет;
- B) да;
- C) иногда.

147. Разрешается ли использование циклов при структурном программировании:

Варианты ответа:

- A) да;
- B) нет.

148. Разрешается ли использование оператора IF при структурном программировании:

Варианты ответа:

- A) да;
- B) нет.

149. Программирование без GO TO применяется. при:

Варианты ответа:

- A) структурном программировании;
- B) модульном программировании;
- C) объектно-ориентированном программировании;
- D) все ответы верные.

150. Достоинство структурного программирования:

Варианты ответа:

- A) можно приступить к комплексному тестированию на раннем этапе разработки;
- B) можно приступить к автономному тестированию на раннем этапе разработки;
- C) нет необходимости выполнять тестирование;
- D) можно пренебречь отладкой.

151. Достоинство структурного программирования:

Варианты ответа:

- A) облегчает работу над большими и сложными проектами;
- B) повышает быстродействие программы;
- C) снижает затраты на программирование.

152. Недостаток структурного программирования:

Варианты ответа:

- A) увеличивает размер программы;
- B) снижает эффективность;
- C) уменьшает количество ошибок;
- D) не требует отладки.

153. Повышает ли читабельность программ структурное кодирование:

Варианты ответа:

- A) да;
- B) нет.

154. Разрешается ли использование циклов при объектно-ориентированном программировании:

Варианты ответа:

- A) да;
- B) нет.

155. Разрешается ли использование оператора IF при объектно-ориентированном программировании:

Варианты ответа:

А) да;

В) нет.

156. Предусматривает ли объектно-ориентированное программирование использование стандартных процедур и функций:

А) да;

В) нет.

157. Можно ли сочетать объектно-ориентированное и структурное программирование

Варианты ответа:

А) можно;

В) нельзя.

158) Можно ли сочетать объектно-ориентированное и модульное программирование:

Варианты ответа:

А) можно;

В) нельзя.

159. Что такое объект, в объектно-ориентированном программировании:

Варианты ответа:

А) тип данных;

В) структура данных;

С) событие;

Д) обработка событий;

Е) использование стандартных процедур.

160. Инкапсуляция это:

Варианты ответа:

А) определение новых типов данных;

В) определение новых структур данных;

С) объединение переменных, процедур и функций в одно целое;

Д) разделение переменных, процедур и функций;

Е) применение стандартных процедур и функций.

161. Наследование это:

Варианты ответа:

А) передача свойств экземплярам;

В) передача свойств предкам;

С) передача свойств потомкам;

Д) передача событий потомкам.

162. Полиморфизм это:

Варианты ответа:

А) изменение поведения потомков, имеющих общих предков;

В) передача свойств по наследству;

С) изменение поведения потомков на разные события;

Д) изменение поведения экземпляров, имеющих общих предков;

163. Три "кита" объектно-ориентированного метода программирования:

Варианты ответа:

А) предки, родители, потомки;

В) полиморфизм, инкапсуляция, наследование;

С) свойства, события, методы;

Д) визуальные, не визуальные компоненты и запросы.

164. Какое утверждение верно:

Варианты ответа:

- A) предки наследуют свойства родителей;
- B) родители наследуют свойства потомков;
- C) потомки не могут иметь общих предков;
- D) потомки наследуют свойства родителей.

165. Может ли дочерний элемент иметь двух родителей:

Варианты ответа:

- A) да;
- B) нет;
- C) только для визуальных элементов;
- D) если их свойства совпадают.

165. Могут ли два визуальных компонента иметь общего предка:

Варианты ответа:

- A) да;
- B) нет;
- C) если их свойства совпадают;
- D) если их методы совпадают.

167. Есть ли различие между объектом и экземпляром:

Варианты ответа:

- A) да;
- B) нет;
- C) если у них общий предок.

168. Есть ли различие в поведении объекта и экземпляра того же типа:

Варианты ответа:

- A) да;
- B) если у них есть общий предок;
- C) нет;
- D) если у них нет общего предков.

169. Изменение свойств, приводит к изменению поведения экземпляра:

Варианты ответа:

- A) нет;
- B) только для визуальных;
- C) только НЕ для визуальных ;
- D) да .

170. Можно ли свойствам присваивать значения:

Варианты ответа:

- A) да (всегда);
- B) не всегда;
- C) нет.

171. Можно ли переопределять методы:

Варианты ответа:

- A) да;
- B) нет.

172. Можно ли переопределять свойства:

Варианты ответа:

- A) да;
- B) нет.

173. Могут ли два различных объекта реагировать на событие по-разному:

Варианты ответа:

- A) да;
- B) нет.

174. Могут ли два экземпляра одного объекта реагировать на событие по-разному:

Варианты ответа:

- A) да;
- B) нет.

175. Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

- A) сверху вниз;
- A) снизу-вверх.

176. Какой этап проектирования может быть исключен:

Варианты ответа:

- A) эскизный проект;
- B) технический проект;
- C) рабочий проект.

177. Какие этапы проектирования можно объединять:

Варианты ответа:

- A) технический и рабочий;
- B) эскизный и рабочий;
- C) технический и эскизный.

178. Модульное программирование применимо при:

Варианты ответа:

- A) проектировании сверху вниз;
- B) проектирование снизу-вверх;

179. Процесс преобразования постановки задачи в план алгоритмического или вычислительного решения это:

Варианты ответа:

- A) проектирование;
- B) анализ требований;
- C) программирование;
- D) тестирование.

180. Составление спецификаций это:

Варианты ответа:

- A) формализация задачи;
- B) эскизный проект;
- C) поиск алгоритма;
- D) отладка.

181. Этап разработки программы, на котором дается характеристика области применения программы:

Варианты ответа:

- A) техническое задание;
- B) эскизный проект;
- C) технический проект;
- D) внедрение;

Е) рабочий проект.

182. Укажите правильную последовательность создания программы:

Варианты ответа:

А) формулирование задачи, анализ требований, проектирование, программирование;

В) анализ требований, проектирование, программирование, тестирование, отладка;

С) анализ требований, программирование, проектирование, тестирование;

Д) анализ требований, проектирование, программирование, модификация, трассировка;

Е) формулирование задачи, анализ требований, программирование, проектирование, отладка.

183. Уточнение структуры входных и выходных данных, разработка алгоритмов, определение элементов интерфейса входят в:

Варианты ответа:

А) технический проект;

В) рабочий проект;

С) эскизный проект.

184. Несуществующий метод проектирования:

Варианты ответа:

А) алгоритмическое;

В) нисходящее;

С) восходящее.

185. Метод проектирования:

Варианты ответа:

А) нисходящее;

В) алгоритмическое;

С) логическое;

Д) использование языков программирования;

Е) составление блок-схем.

186. Нисходящее проектирование это:

Варианты ответа:

А) последовательное уточнение (детализация);

В) составление блок-схем;

С) разделение программы на отдельные участки (блоки);

Д) трассировка.

187. Признаки нисходящего программирования:

Варианты ответа:

А) последовательная детализация;

В) наличие оптимизации;

С) наличие тестирования;

Д) автоматизация программирования.

188. Какой методикой пользуются при структурном программировании:

Варианты ответа:

А) сверху вниз;

В) снизу-вверх.

189. Проектирование сверху вниз это:

Варианты ответа:

А) последовательное разбиение общих задач на более мелкие;

В) составление из отдельных модулей большой программы.

190. Проектирование снизу-вверх это:

Варианты ответа:

А) составление из отдельных модулей большой программы;

В) последовательное разбиение общих задач на более мелкие.

191. Модульное программирование применимо при:

Варианты ответа:

- A) проектировании сверху вниз;
- B) проектирование снизу-вверх;
- C) и в том, и другом случае;
- D) ни в коем случае.

192. Какой методикой проектирования пользуются при структурном программировании:

Варианты ответа:

- A) сверху вниз;
- B) снизу-вверх.

193. В чем заключается иерархический подход в решении задачи:

Варианты ответа:

- A) в последовательном разбиении задачи на более мелкие составные части;
- B) в выделении основных и второстепенных элементов;
- C) в возможности параллельного выполнения отдельных частей задачи.

194. Какой метод проектирования соответствует иерархическому подходу в решении задачи:

Варианты ответа:

- A) нисходящее (сверху вниз);
- B) восходящее (снизу-вверх).

195. В каких единицах измеряются затраты на проектирование:

Варианты ответа:

- A) в человеко-днях;
- B) в долларах;
- C) в тенге;
- D) в килобайтах.

196. Зависит ли трудоемкость разработки от сложности алгоритма:

Варианты ответа:

- A) да;
- B) нет.

197. Зависит ли трудоемкость разработки от количества программистов:

Варианты ответа:

- A) да;
- B) нет.

198. Зависит ли трудоемкость разработки от языка или системы программирования:

Варианты ответа:

- A) да;
- B) нет.

199. Зависит ли трудоемкость разработки от количества обрабатываемой информации:

Варианты ответа:

- A) да;
- B) нет.

200. Зависит ли трудоемкость разработки от вида информации:

Варианты ответа:

- A) да;
- B) нет.

201. Если вы приобрели программу законным путем, являетесь ли вы собственником программы:

Варианты ответа:

А) нет;

В) да.

202. Если вы приобрели программы законным путем, имеете ли вы право вносить в нее изменения:

Варианты ответа:

А) нет;

В) да

203. Если вы приобрели программы законным путем, имеете ли вы право продать ее:

Варианты ответа:

А) да;

В) нет.

204. Кому принадлежит право собственности на ПО:

Варианты ответа:

А) разработчику;

А) продавцу;

В) покупателю.

205. Кому принадлежит авторское право на ПО:

Варианты ответа:

А) разработчику;

С) продавцу;

Д) покупателю.

206. Что охраняется законом:

Варианты ответа:

А) структура базы данных;

В) содержание базы данны

Критерии теста

Каждое тестовое задание имеет определенный порядковый номер, из которых - один верный и три неверных ответа.

Критерии оценивания:

«отлично» - 90%-100% правильных ответов,

«хорошо»- 75%-89% правильных ответов,

«удовлетворительно»- 50%-74% правильных ответов,

«неудовлетворительно»- менее 50% правильных ответов.

Время, которое отводится на выполнение теста- 90мин

4.2. Примерный перечень тестов к экзамену

МДК 02.02 Инструментальные средства разработки программного обеспечения (семестр 4)

1.	Технология ASP.NET	—	1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера; 2) работает на небольших устройствах и позволяет вам создавать приложения для этих
----	--------------------	---	--

		<p>мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
2.	Технология NetCF — .NET Compact Framework 3.5	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
3.	Технология Silverlight	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием HTML, Ajax и обработки на стороне сервера;</p> <p>2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;</p> <p>3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac);</p> <p>4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);</p>
4.	Технология VSTO — Visual Studio Tools for Office	<p>1) позволяет создавать Web-решения (а также решения на базе браузеров) с использованием</p>

		<p>HTML, Ajax и обработки на стороне сервера;</p> <ol style="list-style-type: none"> 2) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств; 3) для разработки высоко интерактивных решений, которые сочетают видео и анимацию (предоставляемые через Интернет для компьютеров под управлением как Windows, так и Mac); 4) позволяет создавать решения на базе инструментов пакета Office (в том числе Outlook и SharePoint);
5.	Технология WinForms —	<ol style="list-style-type: none"> 1) используются для разработки бизнес-приложений и различных инструментов для платформы Windows. 2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows. 3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio. 4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;
6.	Технология WPF Windows Presentation Foundation	<ol style="list-style-type: none"> 1) используются для разработки бизнес-приложений и различных инструментов для платформы Windows. 2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows. 3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio. 4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;
7.	Технология XNA	<ol style="list-style-type: none"> 1) используются для разработки бизнес-приложений и различных инструментов для

		<p>платформы Windows.</p> <ol style="list-style-type: none"> 2) использует WinForms, XAML, Smart Clients, трехмерную графику и прочее для того, чтобы вы могли создавать функционально богатые клиентские решения под Windows. 3) эта технология позволяет вам создавать игры для Xbox при помощи Visual Studio. 4) работает на небольших устройствах и позволяет вам создавать приложения для этих мобильных устройств;
8.	Элемент управления пользовательского интерфейса для Web-разработчика: ListView -	<ol style="list-style-type: none"> 1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами. 2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения. 3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным. 4) поддерживать синхронизированное состояние исходного кода и визуального конструктора.
9.	Элемент управления пользовательского интерфейса для Web-разработчика: DataPager -	<ol style="list-style-type: none"> 1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами. 2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения. 3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным. 4) поддерживать синхронизированное состояние

		исходного кода и визуального конструктора.
10.	Элемент управления пользовательского интерфейса для Web-разработчика: LinqDataSourceControl -	<ol style="list-style-type: none"> 1) упрощает отображение повторяющихся данных. Элемент управления управляется шаблонами. 2) позволяет вам управлять разбиением данных на страницы, а также пользовательским интерфейсом такого разбиения. 3) это новая модель программирования, введенная в Visual Studio 2008. Она сочетает запросы к базе данных с языком программирования .NET. Таким образом, вы можете писать строго типизированный код (а не простые строчки) для запросов к вашим данным. 4) поддерживать синхронизированное состояние исходного кода и визуального конструктора.
11.	Представление Source в Visual Studio 2008	<ol style="list-style-type: none"> 1) дает вам полный доступ к редактированию кода XHTML данной страницы. 2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого вызова, связанным с элементами управления в визуальном конструкторе. 3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.
12.	Представление Design в Visual Studio 2008	<ol style="list-style-type: none"> 1) дает вам полный доступ к редактированию кода XHTML данной страницы. 2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого вызова, связанным с элементами управления в визуальном конструкторе. 3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.
13.	Представление Split в Visual Studio 2008	<ol style="list-style-type: none"> 1) дает вам полный доступ к редактированию кода XHTML данной страницы. 2) позволяет увидеть страницу в ее развитии и дает доступ ко многим командам быстрого

		<p>вызова, связанным с элементами управления в визуальном конструкторе.</p> <p>3) старается поддерживать синхронизированное состояние исходного кода и визуального конструктора.</p>
14.	Visual Studio Team System Development Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
15.	Visual Studio Team System Database Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт</p>

		<p>улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
16.	Visual Studio Team System Architect Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
17.	Visual Studio Team System Test Edition	<p>1) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>2) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p>

		<p>3) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p> <p>4) — предназначена для тестировщика программного обеспечения; эта редакция включает возможность создавать модульные тесты и видеть анализ покрытия кода.</p>
18.	Visual Studio Team Suite	<p>1) — для тех, кому нужно иметь все сразу (и у кого нет бюджетных ограничений) этот продукт содержит в себе абсолютно все.</p> <p>2) — предназначена для большинства разработчиков, в эту версию входят: статический анализ кода, профилирование кода, динамический анализ кода, метрики кода, политики анализа кода, модульное тестирование и анализ покрытия кода.</p> <p>3) — предназначена для разработчиков, которые плотно работают с разработкой баз данных; эта редакция позволяет вам создавать проекты баз данных, генерировать образцовые данные, сравнивать схемы, сравнивать данные, выполнять модульное тестирование на уровне базы данных и т. д.</p> <p>4) — предназначена для архитекторов программного обеспечения; этот продукт улучшает проектирование и проверку правильности проектирования распределенных систем.</p>
19.	Toolbox	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм.</p> <p>2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение.</p> <p>3) окно свойств.</p> <p>4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.</p>
20.	Solution Explorer	<p>1) предоставляет доступ ко множеству элементов управления при создании Web- и</p>

		<p>Windows-форм.</p> <ol style="list-style-type: none"> 2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение. 3) окно свойств. 4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.
21.	Properties	<ol style="list-style-type: none"> 1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм. 2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение. 3) окно свойств. 4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.
22.	Object Browser	<ol style="list-style-type: none"> 1) предоставляет доступ ко множеству элементов управления при создании Web- и Windows-форм. 2) позволяет вам группировать и управлять множеством файлов, которые составляют ваше приложение. 3) окно свойств. 4) отображает иерархическое представление проектов, сборок, пространств имен, типов, перечислений и интерфейсов.
23.	ASP.NET Web Site	<ol style="list-style-type: none"> 1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config. 2) — представляет собой сайт для создания Web-сервиса на основе XML. 3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов. 4) — используется для создания Web-сайта,

		который содержит сервисы WCF.
24.	ASP.NET Web Service	<ol style="list-style-type: none"> 1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config. 2) — представляет собой сайт для создания Web-сервиса на основе XML. 3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов. 4) — используется для создания Web-сайта, который содержит сервисы WCF.
25.	Empty Web Site	<ol style="list-style-type: none"> 1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config. 2) — представляет собой сайт для создания Web-сервиса на основе XML. 3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов. 4) — используется для создания Web-сайта, который содержит сервисы WCF.
26.	WCF Service	<ol style="list-style-type: none"> 1) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога AppData, Web-формы Default.aspx и файла web.config. 2) — представляет собой сайт для создания Web-сервиса на основе XML. 3) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов. 4) — используется для создания Web-сайта, который содержит сервисы WCF.
27.	ASP.NET Reports Web Site	<ol style="list-style-type: none"> 1) — создает сайт для формирования Web-отчетов (при помощи инструментов Microsoft Report Viewer). 2) — представляет собой стандартный Web-сайт ASP.NET, который начинается с каталога

		<p>AppData, Web-формы Default.aspx и файла web.config.</p> <p>3) — представляет собой сайт для создания Web-сервиса на основе XML.</p> <p>4) — представляет собой проект Web-сайта, лишенный всех каталогов и файлов.</p>
28.	Расширение файла aspx	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
29.	Расширение файла asmx	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
30.	Расширение файла cs, vb	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
31.	Расширение файла config	<p>1) Определяет Web-форму ASP NET.</p> <p>2) Определяет Web-сервис на основе XML в ASP.NET.</p> <p>3) Определяет файл класса.</p> <p>4) Представляет собой конфигурационный файл вашего Web-приложения.</p>
32.	Page Prelnit	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически,</p>

		<p>настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p> <ol style="list-style-type: none"> 2) это событие вызывается для создания элементов управления, используемых Web-страницей. 3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления). 4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер. 5) это последнее событие, которое вызывается для страницы (после визуализации страницы).
33.	Page Init—	<ol style="list-style-type: none"> 1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы. 2) это событие вызывается для создания элементов управления, используемых Web-страницей. 3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления). 4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.
34.	Page Load —	<ol style="list-style-type: none"> 1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы. 2) это событие вызывается для создания элементов управления, используемых Web-страницей. 3) это событие вызывается тогда, когда страница загружена (после инициализации элементов

		<p>управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p>
35.	Page Pre Render—	<p>1) это первое событие, для которого вы можете писать код. Полезно в том случае, когда вы создаете элементы управления динамически, настраиваете динамически главную страницу или динамически настраиваете тему страницы.</p> <p>2) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>3) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>4) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p>
36.	Page Unload—	<p>1) это событие вызывается для создания элементов управления, используемых Web-страницей.</p> <p>2) это событие вызывается тогда, когда страница загружена (после инициализации элементов управления).</p> <p>3) это событие вызывается непосредственно перед тем, как окончательная визуализация страницы посылается обратно в браузер.</p> <p>4) это последнее событие, которое вызывается для страницы (после визуализации страницы).</p>
37.	Элемент управления Label	<p>1) для передачи пользователю текстовой информации.</p> <p>2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные.</p> <p>3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некоего действия (такого как сохранение, сброс или отмена).</p>

		4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.
38.	Элемент управления TextBox	<ol style="list-style-type: none"> 1) для передачи пользователю текстовой информации. 2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные. 3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некоего действия (такого как сохранение, сброс или отмена). 4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.
39.	Элемент управления Button	<ol style="list-style-type: none"> 1) для передачи пользователю текстовой информации. 2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные. 3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некоего действия (такого как сохранение, сброс или отмена). 4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.
40.	Элемент управления HyperLink	<ol style="list-style-type: none"> 1) для передачи пользователю текстовой информации. 2) для того, чтобы пользователь мог ввести на вашей странице текстовые данные. 3) используйте этот элемент управления тогда, когда вы ожидаете от пользователя запуска на странице некоего действия (такого как сохранение, сброс или отмена). 4) используйте этот элемент управления тогда, когда вам нужно работать с гиперссылкой на стадии выполнения.
41.	Элемент управления	1) Используйте этот элемент управления для

	DropDownList	<p>предоставления списка опций, из которого пользователь может выбирать.</p> <ol style="list-style-type: none"> 2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено). 3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один. 4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства imageUrl).
42.	Элемент управления CheckBox	<ol style="list-style-type: none"> 1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать. 2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено). 3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один. 4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства imageUrl).
43.	Элемент управления RadioButton	<ol style="list-style-type: none"> 1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать. 2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено). 3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.

		<p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства <code>ImageUrl</code>).</p>
44.	Элемент управления <code>Image</code>	<p>1) Используйте этот элемент управления для предоставления списка опций, из которого пользователь может выбирать.</p> <p>2) Используйте этот элемент управления тогда, когда вы хотите позволить пользователю указывать значение (да или нет, включено или выключено).</p> <p>3) Используйте этот элемент управления для предоставления пользователю группы элементов, из которой можно выбрать только один.</p> <p>4) Используйте этот элемент управления тогда, когда вам нужно динамически заполнять изображение на вашей странице (при помощи свойства <code>ImageUrl</code>).</p>
45.	Элемент управления <code>Table</code>	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
46.	Элемент управления <code>Calendar</code>	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю</p>

		<p>видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
47.	Элемент управления MultiView	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>
48.	Элемент управления Wizard	<p>1) Используйте этот элемент управления для динамического построения таблиц в вашем коде.</p> <p>2) Применяйте этот элемент управления тогда, когда вам нужно позволить пользователю видеть набор дат и выбрать одну из них.</p> <p>3) Используйте этот элемент управления тогда, когда вы хотите обеспечить пользователю несколько представлений (в зависимости от выбора пользователя или от состояния системы).</p> <p>4) Предоставляет дополнительные возможности, необходимые для мастера, который переходит с шага на шаг (или с представления на представление) и обратно.</p>

49.	Элемент управления RangeValidator	<ol style="list-style-type: none"> 1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений. 2) Используйте этот элемент для проверки шаблона введенных пользователем данных. 3) Используйте этот элемент для сравнения ввода пользователя. 4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.
50.	Элемент управления RegularExpressionValidator	<ol style="list-style-type: none"> 1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений. 2) Используйте этот элемент для проверки шаблона введенных пользователем данных. 3) Используйте этот элемент для сравнения ввода пользователя. 4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.
51.	Элемент управления CompareValidator	<ol style="list-style-type: none"> 1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений. 2) Используйте этот элемент для проверки шаблона введенных пользователем данных. 3) Используйте этот элемент для сравнения ввода пользователя. 4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.
52.	Элемент управления CustomValidator	<ol style="list-style-type: none"> 1) Используйте этот элемент для проверки ввода пользователя на определенный диапазон значений. 2) Используйте этот элемент для проверки шаблона введенных пользователем данных. 3) Используйте этот элемент для сравнения

		<p>ввода пользователя.</p> <p>4) Используйте этот элемент для создания вашего собственного проверяющего элемента управления.</p>
53.	Элемент управления Login	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
54.	Элемент управления PasswordRecovery	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
55.	Элемент управления CreateUserWizard	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить</p>

		<p>пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
56.	Элемент управления ChangePassword	<p>1) Этот элемент управления обеспечивает основной интерфейс для запроса у пользователя его учетных данных (имени пользователя и пароля).</p> <p>2) Этот элемент управления используется для восстановления пароля Пользователя.</p> <p>3) Этот элемент управления позволяет пользователям создавать собственные учетные записи или помогает восстановить пароль.</p> <p>4) Этот элемент управления позволяет пользователям вводить текущий пароль и новый пароль.</p>
57.	Элемент управления GridView	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
58.	Элемент управления DataList	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам</p>

		<p>отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
59.	Элемент управления DetailsView	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
60.	Элемент управления Form View	<p>1) Этот элемент управления предназначен для привязки к табличным данным и работы с ними.</p> <p>2) Вы можете использовать этот элемент управления тогда, когда хотите управлять тем, как ваши данные отображаются и форматируются.</p> <p>3) Этот элемент управления позволяет вам отображать одну строку данных.</p> <p>4) Этот элемент управления с дополнительными возможностями. Можно определить шаблоны, в соответствии с которыми будет отображаться данная строка.</p>
61.	Элемент управления ListView	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и</p>

		<p>связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
62.	Элемент управления Repeater	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
63.	Элемент управления DataPager	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов select, update, new и delete.</p>
64.	Элемент управления Data Source Controls	<p>1) вы можете настроить поведение вашего пользовательского интерфейса во время таких операций, как- просмотр, редактирование, добавление и удаление.</p> <p>2) Этот элемент управления является</p>

		<p>контейнером для повторяющихся данных.</p> <p>3) Этот элемент управления позволяет вам управлять разбиением данных на страницы и связанным с этим разбиением интерфейсом пользователя.</p> <p>4) Эти элементы могут быть настроены для работы с данными источников и для выполнения методов <code>select</code>, <code>update</code>, <code>new</code> и <code>delete</code>.</p>
65.	Позиционирование окна формы на экране свойство - <code>CenterParent</code>	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию <code>Windows</code>, границы формы также определяются умолчаниями <code>Windows</code>.</p>
66.	Позиционирование окна формы на экране свойство - <code>CenterScreen</code>	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию <code>Windows</code>, границы формы также определяются умолчаниями <code>Windows</code>.</p>
67.	Позиционирование окна формы на экране свойство - <code>Manual</code>	<p>1) Центрирует форму внутри ее родительской формы.</p> <p>2) Центрирует форму внутри текущего экрана дисплея.</p> <p>3) Форма позиционируется в соответствии со значением свойства <code>Form.Location</code>.</p> <p>4) Позиционирует форму в положение по умолчанию <code>Windows</code>, границы формы также определяются умолчаниями <code>Windows</code>.</p>

68.	Позиционирование окна формы на экране свойство - WindowsDefaultBounds	<ol style="list-style-type: none"> 1) Центрирует форму внутри ее родительской формы. 2) Центрирует форму внутри текущего экрана дисплея. 3) Форма позиционируется в соответствии со значением свойства Form.Location. 4) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows.
69.	Позиционирование окна формы на экране свойство - WindowsDefaultLocation	<ol style="list-style-type: none"> 1) Центрирует форму внутри текущего экрана дисплея. 2) Форма позиционируется в соответствии со значением свойства Form.Location. 3) Позиционирует форму в положение по умолчанию Windows, границы формы также определяются умолчаниями Windows. 4) Позиционирует форму в положение по умолчанию Windows, размер формы определяется свойством Form. Size (это настройка по умолчанию).
70.	Элемент управления TableLayoutPanel	<ol style="list-style-type: none"> 1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров. 2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости. 3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей. 4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.
71.	Элемент управления FlowLayoutPanel	<ol style="list-style-type: none"> 1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов

		<p>на основе Web-браузеров.</p> <ol style="list-style-type: none"> 2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости. 3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей. 4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.
72.	Элемент управления SplitContainer	<ol style="list-style-type: none"> 1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров. 2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости. 3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей. 4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.
73.	Элемент управления ToolStripContainer	<ol style="list-style-type: none"> 1) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров. 2) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости. 3) Представляет собой союз двух панелей и разделителя; разделитель разделяет две

		<p>панели либо по горизонтали, либо по вертикали и позволяет пользователю вручную настроить ту часть пространства внутри общего контейнера, которую занимает каждая из панелей.</p> <p>4) Он является комбинацией четырех панелей, каждая из которых размещается на разных сторонах содержащей их формы.</p>
74.	Элемент управления MenuStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
75.	Элемент управления ToolStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
76.	Элемент управления StatusStrip	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p> <p>4) Элемент управления для представления данных с иерархическими связями.</p>
77.	Элемент управления TreeView	<p>1) Даёт вам возможность визуально конструировать систему главного меню формы.</p> <p>2) Создание панели инструментов.</p> <p>3) Создание строки состояния.</p>

		4) Элемент управления для представления данных с иерархическими связями.
78.	Элемент управления ImageList	<ol style="list-style-type: none"> 1) Работают как провайдер изображений для других элементов управления. 2) Для отображения данных в табличном формате. 3) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров. 4) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.
79.	Элемент управления DataGridView	<ol style="list-style-type: none"> 1) Работают как провайдер изображений для других элементов управления. 2) Для отображения данных в табличном формате. 3) Дает вам такие же возможности компоновки, какие предоставляет HTML для интерфейсов на основе Web-браузеров. 4) Элементы упорядочиваются по вертикали или горизонтали простым переносом элементов управления по рядам или столбцам по мере необходимости.
80.	Базовый класс WPF: visual—	<ol style="list-style-type: none"> 1) этот класс является основным модулем визуализации в WPF. 2) этот класс реализует модель многопоточности в WPF. 3) это базовый класс для элементов управления в WPF. 4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).
81.	Базовый класс WPF: DispatcherObject —	<ol style="list-style-type: none"> 1) этот класс является основным модулем визуализации в WPF.

		<ul style="list-style-type: none"> 2) этот класс реализует модель многопоточности в WPF. 3) это базовый класс для элементов управления в WPF. 4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).
82.	Базовый класс WPF: Control —	<ul style="list-style-type: none"> 1) этот класс является основным модулем визуализации в WPF. 2) этот класс реализует модель многопоточности в WPF. 3) это базовый класс для элементов управления в WPF. 4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).
83.	Базовый класс WPF: Application—	<ul style="list-style-type: none"> 1) этот класс является основным модулем визуализации в WPF. 2) этот класс реализует модель многопоточности в WPF. 3) это базовый класс для элементов управления в WPF. 4) обеспечивает сервисы жизненного цикла приложения, в том числе Run (для запуска приложения) и Exit (для завершения приложения).
84.	Кнопка Swap в WPF конструкторе	<ul style="list-style-type: none"> 1) меняет местами панель XAML и панель конструирования; 2) разделяет панели по вертикали; 3) разделяет панели по горизонтали; 4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).

85.	Кнопка Vertical Split в WPF конструкторе	<ol style="list-style-type: none"> 1) меняет местами панель XAML и панель конструирования; 2) разделяет панели по вертикали; 3) разделяет панели по горизонтали; 4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).
86.	Кнопка Horizontal Split в WPF конструкторе	<ol style="list-style-type: none"> 1) меняет местами панель XAML и панель конструирования; 2) разделяет панели по вертикали; 3) разделяет панели по горизонтали; 4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).
87.	Кнопка Collapse/Expand Panel в WPF конструкторе	<ol style="list-style-type: none"> 1) меняет местами панель XAML и панель конструирования; 2) разделяет панели по вертикали; 3) разделяет панели по горизонтали; 4) минимизирует или восстанавливает нижнюю или левую панель (в зависимости от того режима просмотра, в котором вы находитесь).
88.	Панели компоновки WPF Canvas	<ol style="list-style-type: none"> 1) Контейнерный элемент управления без встроенной логики компоновки. 2) Панель, которая позволяет выполнять закрепление своих дочерних элементов. 3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам. 4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.
89.	Панели компоновки WPF DockPanel	<ol style="list-style-type: none"> 1) Контейнерный элемент управления без встроенной логики компоновки.

		<ol style="list-style-type: none"> 2) Панель, которая позволяет выполнять закрепление своих дочерних элементов. 3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам. 4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.
90.	Панели компоновки WPF Grid	<ol style="list-style-type: none"> 1) Контейнерный элемент управления без встроенной логики компоновки 2) Панель, которая позволяет выполнять закрепление своих дочерних элементов 3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам. 4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.
91.	Панели компоновки WPF StackPanel	<ol style="list-style-type: none"> 1) Контейнерный элемент управления без встроенной логики компоновки 2) Панель, которая позволяет выполнять закрепление своих дочерних элементов 3) Контейнерный элемент управления, который позволяет дочерним объектам позиционироваться по столбцам и рядам. 4) Контейнерный элемент управления, который реализует для своих дочерних элементов управления размещение в виде вертикальной или горизонтальной стопки.
92.	Элементы управления источниками данных: ObjectDataSource	<ol style="list-style-type: none"> 1) Предоставляет другие классы как источники данных. 2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.

		<p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
93.	Элементы управления источниками данных: SqlDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
94.	Элементы управления источниками данных: AccessDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>
95.	Элементы управления источниками данных: XmlDataSource	<p>1) Предоставляет другие классы как источники данных.</p> <p>2) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB.</p> <p>3) Предоставляет базу данных Microsoft Access как источник данных.</p> <p>4) Предоставляет XML-файл как источник данных.</p>

96.	Элементы управления источниками данных: SiteMapDataSource	<ol style="list-style-type: none"> 1) Предоставляет реляционную базу данных как источник данных. Доступ к базам данных Microsoft SQL Server и Oracle можно получить по "родным" протоколам; поддерживаются также ODBC и OLE DB. 2) Предоставляет базу данных Microsoft Access как источник данных. 3) Предоставляет XML-файл как источник данных. 4) Специальный источник данных, который предоставляет карту сайта ASP NET как источник данных.
97.	Web-сервис	<ol style="list-style-type: none"> 1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные. 2) используется для представления и описания данных независимым от платформы образом. 3) обеспечивает средство локализации в Интернете. 4) служит для описания содержимого Web-сервиса и его Web-методов.
98.	XML (Extensible Markup Language, расширяемый язык разметки)	<ol style="list-style-type: none"> 1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные. 2) используется для представления и описания данных независимым от платформы образом. 3) обеспечивает средство локализации в Интернете. 4) служит для описания содержимого Web-сервиса и его Web-методов.
99.	URI (Uniform Resource Identifier)	<ol style="list-style-type: none"> 1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные. 2) используется для представления и описания данных независимым от платформы образом. 3) обеспечивает средство локализации в

		Интернете. 4) служит для описания содержимого Web-сервиса и его Web-методов.
100.	WSDL (Web Service Description Language, язык описания Web-сервиса)	1) представляет собой цельный набор логики приложения, которая выполняет действия и предоставляет данные. 2) используется для представления и описания данных независимым от платформы образом. 3) обеспечивает средство локализации в Интернете. 4) служит для описания содержимого Web-сервиса и его Web-методов.

Критерии теста

Каждое тестовое задание имеет определенный порядковый номер, из которых - один верный и три неверных ответа.

Критерии оценивания:

«отлично» - 90%-100% правильных ответов,

«хорошо»- 75%-89% правильных ответов,

«удовлетворительно»- 50%-74% правильных ответов,

«неудовлетворительно»- менее 50% правильных ответов.

Время, которое отводится на выполнение теста- 90мин

4.3. Контрольная работа

МДК 02.03 Математическое моделирование

(семестр 6)

1. Математическое моделирование это средство для
 - а) изучения свойств реальных объектов в рамках поставленной задачи
 - б) упрощения поставленной задачи
 - в) поиска физической модели
 - г) принятия решения в рамках поставленной задачи

2. Какой модели быть не может?
 - а) вещественной, физической
 - б) идеальной, физической
 - в) вещественной, математической
 - г) идеальной, математической

3. По поведению математических моделей во времени их разделяют на
 - а) детерминированные и стохастические
 - б) статические и динамические

- в) непрерывные и дискретные
 - г) аналитические и имитационные
4. Как называется замещаемый моделью объект?
- а) копия
 - б) оригинал
 - в) шаблон
 - г) макет
5. Что такое математическая модель?
- а) точное представление реальных объектов, процессов или систем, выраженное в математических терминах и сохраняющее существенные черты оригинала
 - б) точное представление реальных объектов, процессов или систем, выраженное в физических терминах и сохраняющее существенные черты оригинала
 - в) приближенное представление реальных объектов, процессов или систем, выраженное в математических терминах и сохраняющее существенные черты оригинала
 - г) приближенное представление реальных объектов, процессов или систем, выраженное в физических терминах и сохраняющее существенные черты оригинала
6. Какие виды математических моделей получаются при разделении их по принципам построения?
- а) аналитические, имитационные
 - б) детерминированные, стохастические
 - в) стохастические, аналитические
 - г) детерминированные, имитационные
7. На какой язык должна быть "переведена" прикладная задача для ее решения с использованием ЭВМ?
- а) неформальный математический язык
 - б) формальный математический язык
 - в) формальный физический язык
 - г) неформальный физический язык
8. Что такое линейное программирование
- а) это направление математического программирования, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными и линейным критерием
 - б) раздел математического программирования, изучающий подход к решению нелинейных задач оптимизации специальной структуры
 - в) метод оптимизации, приспособленный, к задачам, в которых процесс принятия решения, может быть, разбит на отдельные этапы (шаги)
 - г) это направление математического программирования, в котором целевой функцией или ограничением является нелинейная функция
9. Какой метод относится к методам решения задач линейного программирования
- а) симплекс-метод
 - б) метод множителей Лагранжа
 - в) метод хорд
 - г) метод половинного деления

10. Если в критериальной строке симплексной таблицы нет отрицательный коэффициентов, это означает, что
- а) задача неразрешима
 - б) найден оптимальный план на максимум
 - в) найден оптимальный план на минимум
 - г) задача имеет бесконечно много решений
11. В каком случае задача математического программирования является линейной?
- а) если ее целевая функция линейна
 - б) если ее ограничения линейны
 - в) если ее целевая функция и ограничения линейны
 - г) нет правильного ответа
12. Транспортная задача — это
- а) математическая задача линейного программирования специального вида о поиске оптимального распределения однородных объектов из аккумулятора к приемникам с минимизацией затрат на перемещение
 - б) математическая задача нелинейного программирования специального вида о поиске оптимального распределения однородных объектов из аккумулятора к приемникам с минимизацией затрат на перемещение
 - в) математическая задача дробно-линейного программирования специального вида о поиске оптимального распределения однородных объектов из аккумулятора к приемникам с минимизацией затрат на перемещение.
 - г) нет правильного ответа
13. Транспортная задача линейного программирования называется закрытой, если:
- а) суммарные запасы равны суммарным потребностям
 - б) суммарные запасы больше суммарных потребностей
 - в) суммарные запасы меньше суммарных потребностей
 - г) целевая функция ограничена
14. В соответствии с основной теоремой теории транспортных задач всегда имеет решение
- а) открытая транспортная задача
 - б) закрытая транспортная задача
 - в) транспортная задача с ограничениями типа равенств
 - г) транспортная задача с ограничениями типа неравенств

15. При построении опорного плана транспортной задачи методом северо-западного угла первой подлежит заполнению
- а) клетка, расположенная в левом верхнем углу таблицы планирования
 - б) клетка, расположенная в правом верхнем углу таблицы планирования
 - в) клетка с минимальным значением тарифа
 - г) клетка с максимальным значением тарифа
16. При построении опорного плана транспортной задачи на минимум методом минимального элемента первой подлежит заполнению
- а) клетка, расположенная в левом верхнем углу таблицы планирования
 - б) клетка, расположенная в правом верхнем углу таблицы планирования
 - в) клетка с минимальным значением тарифа
 - г) клетка с максимальным значением тарифа
17. Первым шагом алгоритма метода потенциалов является:
- а) нахождение первого псевдоплана
 - б) нахождение первого условно-оптимального плана
 - в) нахождение первого опорного плана
 - г) нахождение первого базисного решения
18. Теория динамического программирования используется:
- а) для решения задач оптимизации без ограничений
 - б) для решения задач управления многошаговыми процессами
 - в) для решения задач нелинейного программирования
 - г) для решения задач линейного программирования
19. Для решения задачи динамического программирования используется:
- а) принцип оптимальности Беллмана
 - б) принцип максимума Понтрягина
 - в) принцип симметрии
 - г) принцип максимума правдоподобия
20. К задачам динамического программирования относится:
- а) задача планирования замены оборудования

- б) задача о рациионе
- в) транспортная задача линейного программирования
- г) задача о назначениях

21. В методе динамического программирования под управлением понимается

- а) совокупность решений, принимаемых на каждом этапе для влияния на ход развития процесса;
- б) совокупность решений, принимаемых на первом этапе процесса;
- в) совокупность решений, принимаемых на последнем этапе процесса
- г) совокупность решений, принимаемых на предпоследнем этапе процесса

22. При решении задачи динамического программирования строятся:

- а) рекуррентные функциональные уравнения Беллмана
- б) функции Лагранжа
- в) штрафные функции
- г) сечения Гомори

23. Что такое системы массового обслуживания

- а) это такие системы, в которые в случайные моменты времени поступают заявки на обслуживание, при этом поступившие заявки обслуживаются с помощью имеющихся в распоряжении системы каналов обслуживания
- б) это совокупность математических выражений, описывающих входящий поток требований, процесс обслуживания и их взаимосвязь
- в) это такие системы, в которые в определенные моменты времени поступают заявки на обслуживание
- г) нет правильного ответа

24. По наличию очередей системы массового обслуживания делятся на

- а) простые, сложные
- б) открытые, замкнутые
- в) ограниченные СМО, неограниченные СМО
- г) СМО с отказами, СМО с очередью

25. По источнику требований СМО делятся на

- а) простые, сложные
- б) открытые, замкнутые
- в) ограниченные СМО, неограниченные СМО

г) СМО с отказами, СМО с очередью

26. Как называется объект, порождающий заявки в СМО

- а) очередь
- б) диспетчер
- в) генератор заявок
- г) узел обслуживания

27. Из чего состоит узел обслуживания в СМО

- а) из диспетчера и генератора заявок
- б) из конечного числа каналов
- в) из очереди и диспетчера
- г) нет правильного ответа

28. Как называется принцип, в соответствии с которым поступающие на вход обслуживающей системы требования подключаются из очереди к процедуре обслуживания

- а) дисциплина очереди
- б) механизм обслуживания
- в) процедура обслуживания
- г) конфигурация очереди

29. Как называется дисциплина очереди, определяемая следующим правилом: «первым пришел – первый обслуживается»

- а) LIFO
- б) GIFO
- в) FIFO
- г) нет правильно ответа

30. Как называется дисциплина очереди, определяемая следующим правилом: "пришел последним – обслуживается первым"

- а) LIFO
- б) GIFO
- в) FIFO
- г) нет правильно ответа

31. Задача о замене оборудования является задачей

- а) нелинейного программирования
- б) динамического программирования
- в) линейного программирования
- г) целочисленного программирования

32. В процессе динамического программирования раньше всех планируется
- а) первый шаг
 - б) последний шаг
 - в) как сказано в условии задачи
 - г) предпоследний шаг
33. Задача, которая возникает при необходимости максимизации дохода от реализации продукции, производимой некоторой организацией, при этом производство ограничено имеющимися сырьевыми ресурсами, называется
- а) задача коммивояжера
 - б) задача о составлении плана производства
 - в) задача о назначении
 - г) задача о рюкзаке
34. Метод минимального элемента — это
- а) один из комбинаторных методов дискретного программирования, при котором гиперплоскость, определяемая целевой функцией задачи, вдавливается внутрь многогранника планов соответствующей задачи линейного программирования до встречи с ближайшей целочисленной точкой этого многогранника
 - б) один из методов отсечения, с помощью которого решаются задачи целочисленного программирования
 - в) один из группы методов определения первоначального опорного плана транспортной задачи
 - г) один из методов, упрощающий определение исходного опорного плана задачи линейного программирования и симплекс-таблицы
35. Метод потенциалов — это
- а) один из методов проверки опорного плана транспортной задачи на оптимальность
 - б) один из комбинаторных методов дискретного программирования, при котором гиперплоскость, определяемая целевой функцией задачи, вдавливается внутрь многогранника планов соответствующей задачи линейного программирования до встречи с ближайшей целочисленной точкой этого многогранника
 - в) один из методов отсечения, с помощью которого решаются задачи целочисленного программирования
 - г) один из группы методов определения первоначального опорного плана транспортной задачи

36. Метод северо-западного угла это

- а) один из методов проверки опорного плана транспортной задачи на оптимальность
- б) один из комбинаторных методов дискретного программирования, при котором гиперплоскость, определяемая целевой функцией задачи, вдавливается внутрь многогранника планов соответствующей задачи линейного программирования до встречи с ближайшей целочисленной точкой этого многогранника
- в) один из методов отсечения, с помощью которого решаются задачи целочисленного программирования
- г) один из группы методов определения первоначального опорного плана транспортной задачи

37. В задачах динамического программирования шаговое управление должно выбираться

- а) с учетом последствий в будущем
- б) с учетом предшествующих шагов
- в) наилучшим для данного шага
- г) лучше, чем предыдущее

38. Метод динамического программирования применяется для решения

- а) задач, которые нельзя представить в виде последовательности отдельных шагов
- б) многошаговых задач
- в) только задач линейного программирования
- г) задач макроэкономики

39. Принцип оптимальности Беллмана состоит в том, что

- а) каковы бы ни были начальное состояние на любом шаге и управление, выбранное на этом шаге, последующие управления должны выбираться оптимальными относительно состояния, к которому придёт система в конце данного шага
- б) совокупность принимаемых решений обеспечит наибольшую локальную выгоду на каждом шаге процесса
- в) совокупность принимаемых решений обеспечит наибольшую локальную выгоду на последнем шаге процесса
- г) нет правильного ответа

40. Часть математического программирования, задачами которой является нахождение экстремума линейной целевой функции на допустимом множестве значений аргументов называется

- а) линейное программирование
- б) динамическое программирование
- в) квадратичное программирование

- г) дискретное программирование
41. К какому классу моделей можно отнести спичечный коробок, если представить его моделью системного блока ПК при планировании своего рабочего места?
- а) это идеальная, математическая модель
 - б) это вещественная, натурная модель
 - в) это вещественная, физическая модель
 - г) это не является моделью
42. Какая из задач не имеет аналитической модели?
- а) поиск оптимального раскроя листа фанеры
 - б) демодуляция аналогового сигнала
 - в) расчет расхода топлива по заданной формуле
 - г) распознавание текста
43. Какая математическая модель не относится к стохастическим?
- а) идеальный газ
 - б) квантовый осциллятор
 - в) материальная точка
 - г) ни одна из предложенных
44. Материальная точка это не только математическая, но и
- а) натурная модель
 - б) физическая модель
 - в) наглядная модель
 - г) знаковая модель
45. Во время поиска лучшего результата были построены две различные математические модели: эксперимент на ЭВМ, моделирующий систему атомов, и дифференциальная система уравнений, решенная численно, от двух полученных результатов взяли среднеквадратичный. Можно ли считать такой метод моделью?
- а) да, это вещественная, математическая
 - б) да, это идеальная, математическая
 - в) да, это вещественная натурная
 - г) нет
46. Какое максимальное количество моделей одного объекта можно составить?
- а) любое количество
 - б) 1
 - в) 3
 - г) 7
47. Сколько классов моделей существует?
- а) 4
 - б) 2
 - в) 3
 - г) нет правильного ответа

48. Какие модели относятся к классу вещественных моделей?
- а) физические, натурные
 - б) идеальные, физические
 - в) наглядные, идеальные
 - г) натурные, идеальные
49. Какие модели нельзя отнести к классу мысленных моделей?
- а) физические
 - б) натурные
 - в) математические
 - г) наглядные
50. Какие модели входят в состав идеальных математических моделей?
- а) аналитические, функциональные, имитационные, комбинированные
 - б) аналоговые, структурные, геометрические, графические, цифровые и кибернетические
 - в) символы, алфавит, языки программирования, упорядоченная запись, топологическая запись, сетевое представление
 - г) нет правильного ответа
51. В чем заключается построение математической модели?
- а) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно и качественно связь между теми или иными процессами и явлениями, между интересующими специалиста математическими величинами, и факторами, влияющими на конечный результат
 - б) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно связь между теми или иными процессами и явлениями, между интересующими специалиста физическими величинами, и факторами, влияющими на конечный результат
 - в) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно связь между теми или иными процессами и явлениями, между интересующими специалиста математическими величинами, и факторами, влияющими на конечный результат
 - г) в определении связей между теми или иными процессами и явлениями, создании математического аппарата, позволяющего выразить количественно и качественно связь между теми или иными процессами и явлениями, между интересующими специалиста физическими величинами, и факторами, влияющими на конечный результат
52. В зависимости от характера исследуемых реальных процессов и систем, на какие группы могут быть разделены математические модели?
- а) непрерывные, имитационные
 - б) детерминированные, стохастические
 - в) имитационные, детерминированные
 - г) стохастические, имитационные

53. Какие группы математических моделей не являются результатом распределения моделей по их поведению во времени?
- а) статические, динамические
 - б) динамические, изоморфные
 - в) изоморфные, динамические
 - г) непрерывные, изоморфные
54. На какие группы можно разделить математические модели по виду входной информации?
- а) статические, непрерывные
 - б) дискретные, непрерывные
 - в) динамические, непрерывные
 - г) динамические, статические
55. На какие группы можно разделить математические модели по степени их соответствия реальным объектам, процессам или системам?
- а) стохастические, изоморфные
 - б) изоморфные, гомоморфные
 - в) детерминированные, стохастические
 - г) нет правильного ответа
56. Как называется модель, если между ней и реальным объектом, процессом или системой существует полное поэлементное соответствие?
- а) стохастическая
 - б) изоморфная
 - в) детерминированная
 - г) гомоморфная
57. Как называются модели, в которых предполагается отсутствие всяких случайных воздействий и их элементы (элементы модели) достаточно точно установлены?
- а) статические
 - б) дискретные
 - в) детерминированные
 - г) динамические
58. В каком моделировании функционирование объектов, процессов или систем описывается набором алгоритмов?
- а) аппроксимационном
 - б) имитационном
 - в) аналитическом
 - г) нет правильного ответа
59. Какие характеристики объекта, процесса или системы устанавливаются на этапе выбора математической модели?
- а) дискретность, изоморфность
 - б) линейность, стационарность
 - в) изоморфность, линейность
 - г) стационарность, дискретность

60. Посредством каких конструкций, математические модели описывают основные свойства объекта, процесса или системы, его параметры, внутренние и внешние связи?
- а) логико-математических конструкций
 - б) статистических конструкций
 - в) вероятностных конструкций
 - г) нет правильного ответа
61. Что не входит в предмет математического моделирования?
- а) построение алгоритма, моделирующего поведение объекта (системы)
 - б) корректировка построенной модели
 - в) поиск закономерностей поведения объекта (системы)
 - г) построение натурной модели
62. Какие изучаются зависимости между величинами, описывающими процессы, при их моделировании?
- а) качественные и количественные
 - б) только качественные
 - в) только количественные
 - г) нет правильного ответа
63. В каких процессах вычислительный эксперимент является единственно возможным?
- а) где натурный эксперимент может привести к очень большим объемам работ
 - б) где натурный эксперимент может привести к неверным результатам
 - в) где натурный эксперимент опасен для жизни и здоровья людей
 - г) нет правильного ответа
64. С чего обычно начинается построение математической модели?
- а) с построения и анализа простейшей, наиболее грубой математической модели рассматриваемого объекта, процесса или системы
 - б) с построения и анализа математической модели, которая наиболее полно соответствует рассматриваемому объекту, процессу или системе
 - в) с анализа математической модели рассматриваемого объекта
 - г) нет правильного ответа
65. Какой характер носят выводы, полученные в результате исследования гипотетической модели?
- а) абстрактный
 - б) условный
 - в) точный
 - г) нет правильного ответа
66. Что необходимо сделать для того, чтобы проверить выводы, полученные в результате исследования гипотетической модели?
- а) необходимо сопоставить результаты исследования модели на ЭВМ с результатами натурного эксперимента

- б) необходимо провести повторное исследование модели и сопоставить результаты двух исследований
- в) необходимо провести исследование модели несколько раз и сопоставить результаты данных исследований
- г) нет правильного ответа

67. При исследовании гипотетической модели какого характера получатся выводы?

- а) абстрактного
- б) условного
- в) гипотетического
- г) динамического

68. Какими знаниями необходимо обладать для построения математической модели в прикладных задачах?

- а) только специальными знаниями об объекте
- б) только математическими знаниями
- в) математическими знаниями и специальными знаниями об объекте
- г) нет правильного ответа

69. Укажите метод, неприменяемый для компьютерного моделирования:

- а) численное решение
- б) точное решение в виде формул
- в) экспериментальный анализ
- г) нет правильного ответа

70. Численный метод предполагает решение в бесконечном цикле итераций. Когда следует прервать процесс вычисления?

- а) в момент, когда решение будет меняться от итерации к итерации менее чем на 1%
- б) когда будет достигнута заданная степень точности
- в) в случае если число начнет расти
- г) нет правильного ответа

71. Какая задача не поддается точному решению на ЭВМ в виде формул?

- а) интегральное уравнение 1-го порядка
- б) дифференциально-интегральная система уравнений
- в) система нелинейных уравнений
- г) все указанные поддаются

72. Какой из методов имеет приближенный характер?

- а) точное решение в виде формул
- б) численное решение
- в) оба указанных метода
- г) нет правильного ответа

73. В чем состоит суть компьютерного моделирования?

- а) на основе математической модели с помощью ЭВМ проводится серия вычислительных экспериментов, т.е. исследуются свойства объектов или процессов, находятся их оптимальные параметры и режимы работы, уточняется модель
- б) в создании математической модели исследуемых объектов
- в) посредством рассмотрения исследуемых объектов с помощью ЭВМ проводится серия вычислительных экспериментов, т.е. исследуются свойства объектов или процессов, находятся их оптимальные параметры и режимы работы, и составляется математическая модель
- г) в создании точной копии исследуемых объектов

74. Какой из экспериментов наиболее выгодно применять для исследования большого числа вариантов проектируемого объекта или процесса для различных режимов его эксплуатации?

- а) прогнозный
- б) вычислительный
- в) натурный
- г) нет правильного ответа

75. Какое преимущество имеет вычислительный эксперимент по сравнению с натурным экспериментом?

- а) короткие сроки и минимальные материальные затраты
- б) только короткие сроки получения результатов
- в) только минимальные материальные затраты
- г) нет правильного ответа

76. Какими методами следует решать системы, состоящие из смешанных (линейных и нелинейных) уравнений?

- а) точными
- б) приближенными
- в) оба предложенных метода годятся
- г) никакими из предложенных

77. Укажите существующие группы решения математических задач

- а) численные, точные
- б) приближенные, точные
- в) численные, приближенные
- г) алгоритмические, приближенные

78. Какие процессы должны отражать математические модели в задачах проектирования или исследования поведения реальных объектов, процессов или систем?

- а) реальные физические нелинейные процессы, протекающие в реальных объектах
- б) реальные математические нелинейные процессы, протекающие в реальных объектах
- в) реальные физические линейные процессы, протекающие в реальных объектах
- г) реальные математические линейные процессы, протекающие в реальных объектах

79. Для чего могут применяться результаты проверки адекватности математической модели и реального объекта, процесса или системы?
- а) только для корректировки математической модели
 - б) только для решения вопроса о применимости построенной математической модели
 - в) для корректировки математической модели или для решения вопроса о применимости построенной математической модели
 - г) нет правильного ответа
80. Что происходит с результатами исследований на ЭВМ при проверке адекватности математической модели и реального объекта, процесса или системы?
- а) сравниваются с результатами эксперимента на опытном натурном образце
 - б) принимаются в качестве итоговых результатов
 - в) не принимаются во внимание
 - г) нет правильного ответа

Критерии контрольной работы

Отметка «5 (отлично)» ставится в случае:

знания, понимания, глубины усвоения обучающимся всего объема программного материала;

творчески применять полученные знания в незнакомой ситуации;

отсутствия ошибок и недочётов при воспроизведении изученного материала, при устных ответах, устранения отдельных неточностей с помощью дополнительных вопросов педагога;

соблюдения культуры письменной и устной речи, правил оформления письменных работ.

Отметка «4 (хорошо)» ставится в случае:

знания всего изученного материала;

умения выделять главные положения в изученном материале, на основании фактов и примеров обобщать, делать выводы, устанавливать межпредметные и внутрипредметные связи, применять полученные знания на практике;

наличие незначительных (негрубых) ошибок при воспроизведении изученного материала;

соблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «3 (удовлетворительно)» ставится в случае:

- знания и усвоения материала на уровне минимальных требований программы, затруднения при самостоятельном воспроизведении, необходимости незначительной помощи учителя;

умения работать на уровне воспроизведения, затруднения при ответах на видоизменённые вопросы;

наличия 1-2 грубых ошибок, нескольких негрубых при воспроизведении изученного материала;

незначительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «2 (неудовлетворительно)» ставится в случае:

знания и усвоения учебного материала на уровне ниже минимальных требований программы;

отсутствия умения работать на уровне воспроизведения, затруднения при ответах на стандартные вопросы;

наличия нескольких грубых ошибок, большого числа негрубых при воспроизведении изученного материала;

- значительного несоблюдения основных правил культуры письменной и устной речи, правил оформления письменных работ.

Отметка «1 (неудовлетворительно)» ставится в случае:

- отказ обучающегося от ответа, выполнения работы, теста, отсутствие выполненного (в том числе, домашнего) задания.

При выставлении отметок необходимо учитывать классификацию ошибок и их количество:

грубые ошибки;

однотипные ошибки;

негрубые ошибки;

недочеты.

К грубым ошибкам следует относить:

незнание определения основных понятий, правил,

неумение выделять главное в ответе;

неумение делать выводы и обобщения;

неумение пользоваться первоисточниками, учебником и справочником.

К однотипным ошибкам относятся ошибки на одно и то же правило.

К негрубым ошибкам следует относить:

неточность формулировок, определений, понятий, правил, вызванная неполнотой охвата основных признаков определяемого понятия или замена 1-2 из этих признаков второстепенными;

нерациональные методы работы с учебной и справочной литературой

Разработчик: Аштуева Анастасия Степановна



ФОС одобрен

на заседании предметно-цикловой комиссии социально-экономических и естественнонаучных дисциплин

Протокол № 8 от «25» марта 2023 г.

Председатель ПЦК



(подпись)

Е.А. Хуснудинова

(Ф.И.О.)