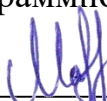


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Малышев Александр Васильевич
Должность: Заведующий кафедрой
Дата подписания: 16.06.2023 12:54:54
Уникальный программный ключ:
с44с65fc5eb466e5e378с4db413465be7586с86f

МИНОБРНАУКИ РОССИИ

Юго-Западный государственный университет

УТВЕРЖДАЮ:
Заведующий кафедрой
программной инженерии


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

«17» июня 2022 г.

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

Системы реального времени
(наименование дисциплины)

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

1 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

1.1 ВОПРОСЫ ДЛЯ СОБЕСЕДОВАНИЯ

Тема 1. Основные понятия технологии программирования.

1. Чем принципиально отличаются процедурное и объектно-ориентированное программирование?
2. Из каких блоков состоит программа, написанная на C++?
3. Чем класс отличается от объектов?
4. Что такое наследование классов и в чем оно состоит?
5. Какие существуют типы наследования?
6. Чем константа отличается от переменной?
7. Что такое оператор инкремента и декремента?
8. Какие логические операторы и операторы сравнения используются в C++?
9. Что такое указатель?
10. Что такое массив?

Тема 2. Методология обеспечения технологичности программных продуктов.

1. Что понимают под технологичностью?
2. Чем определяется технологичность ПО?
3. Когда особенно важна высокая технологичность?
4. Что является результатом процедурной декомпозиции?
5. Что является результатом объектной декомпозиции?
6. Что такое модуль?
7. В каких двух смыслах используют понятие модуль?
8. Первоначальные требования к модулям?
9. Что такое сцепление?
10. Какие типы сцепления существуют?
11. Что такое сцепление по данным?
12. Что такое сцепление по образцу?
13. Что такое сцепление по управлению?
14. Что такое сцепление по общей области данных?
15. Что такое сцепление по содержимому?
16. Что такое связность?

Тема 3. Определение требований к программному обеспечению.

1. Классификация программных продуктов по функциональному признаку.
2. Классификация по назначению.
3. Чем определяют эксплуатационные требования?
4. Что такое правильность?

5. Что такое универсальность?
6. Что такое надежность?
7. Что такое проверяемость?

Тема 4. Анализ требований и определение спецификации программного обеспечения при структурном подходе.

1. Что является результатом анализа требований?
2. Что такое спецификация?
3. Какие бывают спецификации?
4. Что описывают функциональные спецификации?
5. Что определяют эксплуатационные?
6. Какие требования применяются к функциональной спецификации?
7. На какие две группы делятся формальные модели?
8. Какие типы моделей используют в рамках структурного подхода?

Тема 5. Проектирование программного обеспечения при структурном подходе.

1. С чего начинается процесс проектирования?
2. В виде чего может быть представлен результат уточнения структуры?
3. Какую схему называют структурной?
4. Какой самый простой вид ПО?
5. Каким методом выполняют разработку структурной схемы программы?
6. Пример структурной схемы программного комплекса.
7. Пример структурной схемы программной системы.
8. Что такое функциональная схема?
9. Что понимают под проектированием структур данных?
10. Что могут рассматривать в качестве элементов данных?
11. Что представляет собой векторная структура?
12. Что такое списковые структуры?

Тема 6. Структурное программирование на языке высокого уровня.

1. Что такое функция?
2. Как она объявляется?
3. Каким образом функция может возвращать результат?
4. Как функции передаются аргументы?
5. Что такое передача аргумента по значению и по ссылке?
6. Чем эти способы отличаются и как реализуются?
7. Какие аргументы у главного метода программы функции main()?
8. Что такое рекурсия и в каких случаях она используется?
9. Каким образом функции в качестве аргумента передаются массивы?
10. Что такое аргументы по умолчанию и как они определяются?

Тема 7. Проектирование программного обеспечения при объектном подходе.

1. Что является основной задачей логического проектирования?
2. Что включается в себя физическое проектирование при объектном подходе?
3. К каким типам можно отнести большинство классов?
4. Что такое классы-сущности?
5. Что такое граничные классы?
6. Что такое управляющие классы?
7. Что такое класс исключение?
8. Что показывает диаграмма пакетов?
9. Какие виды зависимостей классов есть?
10. Что такое диаграмма кооперации?
11. Что такое агрегация?
12. Что такое композиция?
13. Что такой интерфейс в UML?
14. Чем определяется структура объектов?

Тема 8. Объектно-ориентированное программирование.

1. Как создаются конструкторы?
2. Как создаются деструкторы?
3. Что такое виртуальные методы?
4. Как происходит переопределение виртуальных методов?
5. Что такое многоуровневое наследование?
6. Что такое многократное наследование?
7. Что такое виртуальный метод?
8. Что такое абстрактный класс?
9. Зачем нужно освобождать неиспользуемую динамически выделенную память и как это делается?
10. Как динамически выделяется память для массива?

Тема 9. Разработка пользовательских интерфейсов.

1. Определение интерфейса пользователя?
2. Что такое составные части интерфейса пользователя?
3. Какие основные критерии качества интерфейса?
4. Краткая характеристика критериев качества интерфейса?
5. Что такое модальный и немодальный интерфейс?
6. Зачем нужно обучение пользователей работе с интерфейсом?
7. Какие есть средства обучения пользователей работе с интерфейсом?
8. Назначение и типы меню.
9. Что такое контекстное меню и как оно появилось?
10. Назовите типы окон?
11. Назовите основные элементы окон?

12. Три основных этапа разработки пользовательского интерфейса.

Тема 10. Тестирование программных продуктов.

1. Какие стадии тестирования предполагает процесс разработки?
2. Какие два подхода к формированию тестов есть?
3. Что такое тестирование?
4. Какие основные принципы тестирования существуют?
5. Что такое структурный подход?
6. Что такое функциональный подход?
7. Назовите основные методы ручного контроля.
8. Что такое структурное тестирование?
9. Что такое восходящее тестирование?
10. Что такое нисходящее тестирование?
11. На какие группы можно разделить критерии завершения тестирования?

Тема 11. Отладка программного обеспечения.

1. Что такое отладка?
2. Что такое локализация?
3. Какие ошибки различают?
4. Способы проявления ошибок выполнения.
5. Причины ошибок выполнения.
6. Назовите методы отладки ПО.
7. Что такое метод ручного тестирования?
8. Что такое метод индукции?
9. Что такое метод дедукции?
10. Что такое метод обратного прослеживания?
11. Какие есть методы получения дополнительной информации об ошибке?
12. Что такое отладочный вывод?
13. Что такое интегрирование средств отладки?
14. Что такое отладка с использованием независимых отладчиков?
15. Перечислите этапы отладки.

Тема 12. Составление программной документации.

1. Документация каких типов есть?
2. Какой самый главный документ для разработчика?
3. Что такое ПМИ?
4. Что такое руководство системного программиста?
5. Что такое руководство оператора?
6. Что такое руководство программиста?
7. Какие основные программные документы есть?

Критерии оценки:

1 балл выставляется обучающемуся, если он демонстрирует глубокое знание содержания вопроса; дает точные определения основных понятий; аргументированно и логически стройно излагает учебный материал; иллюстрирует свой ответ актуальными примерами (типовыми и нестандартными), в том числе самостоятельно найденными; не нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

0,5 балла выставляется обучающемуся, если он освоил основные положения контролируемой темы, но недостаточно четко дает определение основных понятий и дефиниций; затрудняется при ответах на дополнительные вопросы; приводит недостаточное количество примеров для иллюстрирования своего ответа; нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

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

1.2 ВОПРОСЫ И ЗАДАНИЯ В ТЕСТОВОЙ ФОРМЕ

Тема 1

Задание №1

Какой язык программирования реализует только объектный подход к программированию?

1. Java
2. JavaScript
3. C++
4. Delphi

Задание №2

Выберите правильную последовательность основных этапов разработки программного обеспечения

1. Постановка задачи, анализ требований и определение спецификаций, проектирование, реализация, сопровождение
2. Постановка задачи, проектирование, анализ требований и определение спецификаций, реализация, сопровождение
3. Постановка задачи, анализ требований и определение спецификаций, проектирование, сопровождение, реализация
4. Последовательность этапов разработки программного обеспечения может быть любой

Задание №3

Какая модель разработки программного обеспечения предусматривает создание программного обеспечения не сразу, а итерационно с использованием метода, базирующегося на создании прототипа?

1. Спиральная модель
2. Каскадная модель
3. Модель с промежуточным контролем
4. Любая модель

Задание №4

Наиболее высокий уровень организации – разработчика программного обеспечения – это:

1. Оптимизирующий уровень
2. Повторяемый уровень
3. Определенный уровень
4. Управляемый уровень

Задание №5

Какие типы сцепления модулей являются допустимыми?

1. По данным, по образцу, по управлению
2. По данным, по образцу, по общей области
3. По управлению, по общей области, по содержимому
4. По образцу, по управлению, по общей области

Задание №6

При промышленном изготовлении сложного программного обеспечения следует использовать:

1. Нисходящий подход.
2. Восходящий подход.
3. Это зависит от того, используется ли в программе управление базой данных.
4. Безразлично, какой подход выбрать: восходящий или нисходящий.

Задание №7

Какой оператор в языке Паскаль не соответствует принципу структурного программирования?

1. Goto
2. If...then...else
3. For
4. While...do

Задание №8

Какая рекомендация позволяет экономить память, а не время выполнения программы

1. Размещать массивы, используемые ограниченное время, в динамической памяти и удалять их при завершении обработки.
2. Выносить вычисление константных, т. е. не зависящих от параметров цикла, выражений из циклов.
3. Избегать операций умножения и деления, заменяя их сложением, вычитанием и сдвигами.
4. Оптимизировать запись условных выражений – исключить лишние проверки.

Тема 2.

Задание №9

Какие программные продукты относят к системным?

1. Операционные системы, оболочки, утилиты.
2. Автоматизированные системы управления реального времени.
3. CASE – средства, среды разработки, отладочные средства.
4. Бухгалтерские системы, издательские системы, профессиональные графические системы.

Задание №10

Документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки называется:

1. Техническим заданием
2. Пояснительной запиской
3. Руководством пользователя
4. Руководством системного программиста

Задание №11

Различие между пакетом программ, программным комплексом и программной системой заключаются в том, что:

1. В пакете программ программы связаны между собой только общей темой, в программном комплексе программой-диспетчером, в программной системе общими данными.

2. В пакете программ программы связаны между собой только общей темой, в программном комплексе общими данными, в программной системе программой-диспетчером.

3. В пакете программ программы связаны между собой программой-диспетчером, в программном комплексе общими данными, в программной системе только общей темой.

4. Между пакетом программ, программным комплексом и программной системой нет никаких существенных различий.

Задание №12

Языки низкого уровня типа Ассемблера

1. Обеспечивают оптимальные с точки зрения времени выполнения и объема памяти программы.

2. Поддерживают принцип структурного программирования.

3. Поддерживают объектный подход к программированию.

4. Используют для разработки сложных программных систем прикладного назначения.

Тема 3.

Задание №13

Какие модели этапа анализа и определения спецификаций относятся к структурному подходу?

1. Диаграммы потоков данных.

2. Диаграммы вариантов использования.

3. Диаграммы последовательностей.

4. Диаграммы деятельности.

Задание №14

Выберите вариант, при котором отношение между сущностями относится к типу «один –ко- многим»

1. Куратор и студент.

2. Студент и номер зачетной книжки.

3. Преподаватель и студент.
4. Преподаватель и учебная дисциплина.

Задание №15

При проектировании программного обеспечения при структурном подходе какие схемы более информативны: структурные или функциональные?

1. Функциональные.
2. Структурные.
3. Это зависит от характера решаемой задачи.
4. Структурные и функциональные схемы – это одно и то же.

Задание №16

Какие структуры данных можно размещать как в статической, так и в динамической памяти ?

1. Векторные структуры.
2. Списковые структуры.
3. Векторные и списковые структуры.
4. Таких структур данных не существует.

Тема 4.

Задание №17

Какие блоки в программе на языке C++ являются обязательными ?

1. Блок заголовков программы и главный метод программы main().
2. Блок заголовков программы и блок с объявлением классов.
3. Блок заголовков программы и блок с описанием функций.
4. Главный метод программы main () и блок с описанием функций.

Задание №18

Чем в языке программирования C++ отличаются операторы while () и do-while () ?

1. Команды цикла while могут не выполниться ни разу, команды цикла do-while выполняются хотя бы 1 раз.
2. В операторах while и do-while нет никаких различий.
3. Команды цикла do-while могут не выполниться ни разу, команды цикла while выполняются хотя бы 1 раз.
4. Команды цикла while выполняются, если условие истинно, команды цикла do-while, если ложно.

Задание №19

В языке C++ элемент двумерного массива $a[1][3]$ находится

1. На пересечении второй строки и четвертого столбца
2. На пересечении четвертой строки и второго столбца
3. На пересечении первой строки и третьего столбца
4. На пересечении третьей строки и первого столбца

Задание №20

Чем в C++ отличаются два механизма передачи аргументов функциям: по значению и через ссылку?

1. При передаче аргументов по значению изменить аргументы в теле функции нельзя, а по ссылке – можно.
2. При передаче аргументов по ссылке изменить аргументы в теле функции нельзя, а по значению – можно.
3. Нет никаких различий.
4. При передаче аргумента по значению функции передается значение переменной, а по ссылке – адрес переменной.

Тема 5.

Задание №21

Какие модели этапа анализа и определения спецификаций относятся к объектному подходу?

1. Диаграммы последовательностей.
2. Функциональные диаграммы.
3. Диаграммы потоков данных.
4. Диаграммы отношений компонентов данных.

Задание №22

Какая модель этапа анализа и определения спецификаций при объектном подходе имеет вид алгоритма?

1. Диаграмма деятельности.
2. Диаграммы вариантов использования.
3. Контекстные диаграммы классов.
4. Диаграммы последовательностей.

Тема 6.

Задание №23

С какой диаграммы следует начинать проектирование программного обеспечения при объектном подходе?

1. Диаграмма пакетов.
2. Уточненная диаграмма классов.
3. Диаграмма последовательностей этапа проектирования.
4. Любая диаграмма.

Задание №24

Какая диаграмма отражает физические взаимосвязи между программными и аппаратными компонентами системы?

1. Диаграмма размещения.
2. Диаграмма компонентов.
3. Диаграмма пакетов.
4. Уточненная диаграмма классов.

Задание №25

Какой член является общим для всех объектов класса?

1. Статический.
2. Открытый.
3. Закрытый.
4. Таких членов класса не существует.

Тема 7.

Задание № 26

По умолчанию члены класса считаются

1. Закрытыми.
2. Открытыми.
3. Защищенными.
4. Статическими.

Задание № 27

Public – член базового класса при механизме Private-наследования в производном классе имеет доступ

1. Private.
2. Public.
3. Protected.
4. Не наследуется.

Задание № 28

В чем различие между перегрузкой методов и переопределением методов?

1. При перегрузке меняется прототип метода, при переопределении прототип метода остается неизменным.
2. При переопределении меняется прототип метода, при перегрузке прототип метода остается неизменным.
3. Нет никаких различий.
4. Перегруженный метод обязательно реализуется только в производном классе, переопределенный метод может быть реализован как в базовом, так и в производном классе.

Тема 8.

Задание № 29

CASE-средство – это

1. Программное средство, поддерживающее процессы жизненного цикла ПО
2. Язык программирования высокого уровня
3. Система управления базами данных
4. Графический редактор

Задание № 30

Erwin – это CASE – средство, предназначенное для :

1. Проектирования баз данных
2. Документирования
3. Тестирования
4. Управления проектом

Задание № 31

Какой программный продукт не является CASE-средством ?

1. MySQL
2. Silverrun
3. Oracle Designer
4. SoDA – Software Document Automation

Задание № 32

Какой этап внедрения CASE-средств является первым?

1. Определение потребностей в CASE – средствах.
2. Оценка и выбор CASE – средств.
3. Выполнение пилотного проекта.
4. Практическое внедрение CASE – средств.

Тема 9.

Задание № 33

Какой интерфейс пользователя является объектно-ориентированным?

1. Прямого манипулирования.
2. Примитивный.
3. Меню.
4. Со свободной навигацией.

Задание № 34

Для какого интерфейса наиболее характерно использование компонентов: кнопки, строки ввода и вывода, списки, реверсивные счетчики, переключатели, флажки ?

1. Со свободной навигацией.
2. Примитивный.
3. Меню.
4. Прямого манипулирования.

Задание № 35

Какая форма диалога интерфейса предполагает использование команд специально разработанного формального языка ?

1. Директивная.
2. Фразовая.
3. Табличная.

4. Любая форма диалога.

Задание № 36

Каким образом тип диалога интерфейса связан с формой диалога интерфейса?

1. Любой тип диалога может быть связан с любой формой диалога.
2. Диалог, управляемый пользователем, может быть связан с фразовой или табличной формой, а диалог, управляемый системой, с директивной формой.
3. Диалог, управляемый пользователем, может быть связан с директивной формой., а диалог, управляемый системой, с фразовой или табличной формой
4. Это зависит от используемого языка программирования и операционной системы.

Тема 10.

Задание № 37

Какой подход к тестированию программного обеспечения не предусматривает знания блок-схемы алгоритма?

1. Функциональное тестирование.
2. Структурное тестирование.
3. Функциональное тестирование и структурное тестирование.
4. Такого подхода к тестированию программного обеспечения не существует.

Задание № 38

Выберите вариант, в котором критерии структурного тестирования расположены в порядке постепенного улучшения.

1. Покрытие операторов, покрытие решений, покрытие условий, покрытие решений / условий, комбинаторное покрытие условий.
2. Покрытие операторов, покрытие условий, покрытие решений, покрытие решений / условий, комбинаторное покрытие условий.
3. Покрытие решений, покрытие условий, покрытие решений / условий, комбинаторное покрытие условий, покрытие операторов.
4. Порядок критериев, обеспечивающий постепенное улучшение, зависит от решаемой задачи.

Задание № 39

Какой критерий структурного тестирования не всегда приводит к выполнению каждого оператора хотя бы 1 раз?

1. Покрытие решений.
2. Покрытие решений / условий.
3. Комбинированное покрытие условий.
4. Покрытие операторов.

Задание № 40

При каком методе функционального тестирования используется математический аппарат алгебры логики?

1. Анализ причинно-следственных связей.
2. Эквивалентное разбиение.
3. Анализ граничных условий.
4. Предположение об ошибке.

Тема 11.

Задание № 41

К какому виду ошибок относится следующая: предусмотрено обращение к подпрограмме другого модуля, но при объединении модулей не стыкуются списки параметров?

1. Ошибка компоновки.
2. Синтаксическая ошибка.
3. Ошибка выполнения.
4. Здесь нет ошибки, программа будет работать правильно.

Задание № 42

При каком методе отладки программного обеспечения вначале формируют множество причин, которые могут вызвать ошибку. Затем анализируют причины и исключают те, которые противоречат имеющимся данным.

1. Метод дедукции.
2. Метод индукции.
3. Метод ручного тестирования.
4. Метод обратного прослеживания.

Задание № 43

Какой метод получения дополнительной информации об ошибке не эффективен при больших и сложных программах?

1. Отладочный вывод.
2. Интегрированные средства отладки.
3. Отладка с использованием независимых отладчиков.
4. Такого метода не существует.

Задание № 44

Выберите правильную последовательность этапов отладки программного обеспечения.

1. Изучение проявления ошибки, локализация ошибки, определение причины ошибки, исправление ошибки, повторное тестирование.
2. Локализация ошибки, изучение проявления ошибки, определение причины ошибки, исправление ошибки, повторное тестирование.
3. Изучение проявления ошибки, определение причины ошибки, локализация ошибки, исправление ошибки, повторное тестирование.
4. Последовательность этапов отладки программного обеспечения зависит от конкретной задачи.

Тема 12.

Задание № 45

Какой программный документ должен содержать сведения для установки программного обеспечения, его настройки и проверки работоспособности ?

1. Руководство системного программиста.
2. Руководство оператора.
3. Руководство программиста.
4. Пояснительная записка

Задание № 46

Какой программный документ должен содержать схемы алгоритмов, а также обоснование принятых технических и технико-экономических решений?

1. Пояснительная записка.
2. Описание программы.
3. Программа и методика испытаний.
4. Ведомость эксплуатационных документов.

Задание № 47

Какой программный документ нельзя объединить с другими документами?

1. Формуляр.
2. Руководство оператора.
3. Руководство программиста.
4. Руководство системного программиста.

Задание № 48

На каком этапе определяется необходимость программных документов (например, текст программы, описание программы, ведомость эксплуатационных документов)?

1. На этапе разработки и утверждения технического задания.
2. На этапе разработки спецификаций программного обеспечения.
3. На этапе проектирования программного обеспечения.
4. На этапе разработки программного кода.

Критерии оценки:

1 балл по шкале БРС выставляется обучающемуся, если даны правильные ответы на 2 вопроса из 2;

0,5 балла по шкале БРС выставляется обучающемуся, если дан правильный ответ на 1 вопрос из 2;

0 баллов по шкале БРС выставляется обучающемуся, если дан правильный ответы на 0 вопросов из 2.

1.3 КОНТРОЛЬНЫЕ ВОПРОСЫ К ЛАБОРАТОРНЫМ РАБОТАМ

Контрольные вопросы к лабораторной работе №1

1. Что такое программный модуль?
2. Какие бывают модули?
3. Что такое объектный модуль?
4. Что такое загрузочный модуль?
5. На какие разделы можно разделить структуру программного модуля?
6. Перечислите основные характеристики программных
7. Что такое размер модуля?
8. Что такое прочность модуля?
9. Что такое прочность модуля?

10. Чем принципиально отличаются процедурное и
11. объектно-ориентированное программирование?
12. Из каких блоков состоит программа, написанная в C++.
13. Какие основные типы данных используются в C++.
14. Чем константа отличается от переменной? Как
15. объявляется константа?
16. Какие арифметические операторы используются в C++.
17. Что такое оператор инкремента и декремента?
18. Каким образом функция может возвращать результат?
19. Как функции передаются аргументы? Что такое передача аргумента по значению и по ссылке? Чем эти способы отличаются и как реализуются?
20. Какие аргументы у главного метода программы–функции `main ()`?
21. Что такое рекурсия и в каких случаях она используется?
22. Каким образом функции в качестве аргумента
23. передаются массивы?

Контрольные вопросы к лабораторной работе №2

1. Что такое адаптивный пользовательский интерфейс?
2. Перечислите преимущества АUI.
3. Перечислите недостатки АUI.
4. Перечислите типы АUI.
5. Какие типы интерфейсов существуют? Дайте их краткую характеристику.
6. К какому типу пользовательского интерфейса можно отнести рабочий стол Windows?
7. Какие модели интерфейсов существуют?
8. Каким образом модели должны быть согласованы между собой?
9. Какие типы диалогов используются при разработке интерфейсов? В чем их основное различие?
10. Какие формы диалогов используются при разработке интерфейсов? В чем их основное различие?
11. Каким образом должны быть согласованы между собой типы диалогов и формы диалогов?
12. Перечислите основные этапы разработки пользовательского интерфейса.
13. Что такое виртуальные методы?
14. Как происходит переопределение виртуальных методов?
15. Что такое многоуровневое наследование?
16. Что такое многократное наследование?
17. Что такое чисто виртуальный метод?

Контрольные вопросы к лабораторной работе №3

1. Что такое интегрированная среда разработки?
2. Что такое СУБД?
3. Назовите основные функции СУБД.
4. Какие бывают СУБД?
5. Назовите наиболее популярные СУБД.
6. Каким образом выполняется чтение и запись текстовых файлов?
7. Каким образом выполняются двоичные операции файлового ввода-вывода?
8. Каким образом осуществляется считывание и запись в файл блоков данных?
9. Каким образом в программе можно обнаружить конец файла?
10. Что такое структура и как она определяется?
11. Что такое поля структуры и как к ним выполняется обращение?
12. Чем переменная (экземпляр) структуры отличается от непосредственно структуры?
13. Каким образом создаются массивы структур?
14. Что такое класс и объект? Чем класс отличается от объекта?

Контрольные вопросы к лабораторной работе №4

1. Как описывается класс и как создаются объекты?
2. Что такое поля и методы класса? Как они объявляются?
3. Чем закрытые члены класса отличаются от открытых?
4. Что такое статическое поле? В чем его особенности?
5. Что такое перегрузка методов и как она выполняется?
6. Что такое конструктор? Когда он вызывается?
7. Что такое деструктор? Как и когда он вызывается?
8. Как создаются конструкторы?
9. Как создаются деструкторы?
10. Что такое наследование классов и в чем оно состоит?
11. Какие существуют типы наследования?

Контрольные вопросы к лабораторной работе №5

1. Какие логические операторы и операторы сравнения используются в C++?
2. Что такое автоматическое приведение типов? Как и когда оно выполняется? Как в C++ выполняется явное приведение типов?

3. Когда и как используются условный оператор `if()`?
4. Каковы его особенности?
5. В чем особенности условного оператора `switch()`? Когда и как он используется?
6. Когда и как используется оператор цикла `for()`? В каких случаях он используется и каковы его особенности?
7. Когда и как используется оператор цикла `while()`? В каких случаях он используется и каковы его особенности?
8. В чем особенности оператора цикла `do-while()` по сравнению с оператором цикла `while()`?
9. Что такое инструкция безусловного перехода? Каков принцип её использования?
10. Что такое абстрактный класс?

Контрольные вопросы к лабораторной работе №6

1. Что такое указатель? Как указатель объявляется и как используется? Какие операции допустимы с указателями?
2. Что такое ссылка? Чем она отличается от указателя?
3. Что такое массив? Как индексируются элементы массива?
4. Выполняется ли в C++ проверка выхода за пределы массива?
5. Как объявляется одномерный массив? Как выполняется обращение к элементам массива?
6. Как объявляются двумерные массивы? Как выполняется обращение к элементам двумерного массива?
7. Как выполняется инициализация массивов?
8. Как имя массива связано с указателем на первый его элемент?
9. В чем особенность массивов, состоящих из символов?
10. Что такое функция? Как она объявляется?

Контрольные вопросы к лабораторной работе №7

1. Что такое API?
2. Перечислите принципы успешного тестирования безопасности.
3. Какие типы тестирования API есть?
4. Какие требования к безопасности системы существуют?
5. Какие языковые и программные средства включают субд?
6. Перечислите файлы проекта ИСР C++Builder и объясните их назначение.
7. Какие компоненты ИСР C++Builder можно использовать

для ввода и вывода текстовой информации?

8. В чем отличие между компонентами CheckBox и RadioButton с точки зрения выбора возможных вариантов?
9. Какие компоненты ИСР C++Builder можно использовать для выбора элементов из списка?
10. Объясните принцип метода Navigate компонента CppWebBrowser.
11. Каким образом следует задать параметры метода Navigate?
12. Что такое аргументы по умолчанию и как они
13. определяются?
14. Каким образом в C++ реализуются текстовые строки?
15. Что такое динамическое выделение памяти и в каких
16. случаях к нему прибегают? Как динамически выделяется память
17. под переменную?
18. Зачем нужно освобождать неиспользуемую динамически
19. выделенную память и как это делается?
20. Как динамически выделяется память для массива?

Критерии оценки:

3-4 балла (или оценка «отлично») выставляется обучающемуся, если он демонстрирует глубокое знание содержания вопроса; дает точные определения основных понятий; аргументированно и логически стройно излагает учебный материал; иллюстрирует свой ответ актуальными примерами (типовыми и нестандартными), в том числе самостоятельно найденными; не нуждается в уточняющих и (или) дополнительных вопросах преподавателя.

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

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

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

2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ

2.1 ТЕМЫ КУРСОВЫХ РАБОТ

1. Система ТО-ДО
2. Система отработки заявок
3. Карточная игра (на выбор)
4. Игра Змейка
5. Игра морской бой
6. Игра 2048
7. Игра Судоку
8. Игра шашки
9. Система отзывов
10. Система шифрования (RSA)
11. Система шифрования (скремблер)
12. Верификатор решения задач
13. Система тестирования
14. Система голосования
15. Система обмена сообщениями
16. Справочник опасных веществ
17. Расчет опасных факторов
18. Система мониторинга курсов валют/новостей/музыки
19. Система формирования отчетной документации на базе шаблонных форм
20. Деревья оптимального поиска.
21. Задача об оптимальной триангуляции многоугольника
22. Алгоритмы полнотекстовой индексации документов
23. Алгоритмы умножения больших чисел

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

Критерии оценивания:

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

сформулированы мотивированные рекомендации; выполнены требования к оформлению курсовой работы.

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

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

0-49 баллов (или оценка «неудовлетворительно») выставляется обучающемуся, если тема курсовой работы не раскрыта и (или) в изложении темы имеются грубые ошибки; структура курсовой работы нечеткая или не определяется вообще; количество изученных источников значительно менее рекомендуемого, неправильно сделаны ссылки на источники или они отсутствуют; не приведены примеры или приведены неверные примеры; отсутствует вывод или автор испытывает затруднения с выводами; не соблюдаются требования к оформлению курсовой работы.

2.2 БАНК ВОПРОСОВ И ЗАДАНИЙ В ТЕСТОВОЙ ФОРМЕ

Задания в закрытой форме

1. Программный продукт – это
 - 1) программа для удовлетворения нужд разработчиков, предназначенная для продажи
 - 2) комплекс взаимосвязанных программ для решения определенной проблемы массового спроса, подготовленный к реализации как любой вид промышленной продукции
 - 3) программная реализация решения задачи на компьютере
 - 4) результат разработки какого-либо технического задания

2. Отличительной особенностью программных продуктов является

1) системность

2) простота

3) универсальность

4) надежность

3. Сопровождение программного продукта – это

1) снабжение программного продукта необходимой документацией

2) обнаружение и исправление ошибок

3) поддержка работоспособности программного продукта, переход на его новые версии, внесение изменений, исправление обнаруженных ошибок и т.д.

4) проверка работоспособности каждой разработанной функции, процедуры, модуля

4. Мобильность программных продуктов – это

1) независимость от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п.

2) точность выполнения предписанных функций обработки

3) способность к внесению изменений

4) обеспечение дружественного интерфейса для работы конечного пользователя, наличие контекстно-зависимой подсказки или обучающей системы в составе программного средства

5. В условиях существования рынка программных продуктов важными его характеристиками являются:

1) количество продаж, наличие программ-конкурентов, длительность продаж

2) стоимость, количество продаж, время нахождения на рынке, известность фирмы-разработчика и программы

3) внешний интерфейс программы, количество продаж, наличие программ-конкурентов

4) модифицируемость, надежность, универсальность, известность фирмы - разработчика

6. Утилитарные программы выполняют роль...

1) сервиса

2) клиента

3) сервера

4) программного средства разработки приложений

7. Основными показателями качества программных продуктов является:

1) алгоритмическая сложность, полнота и системность функций обработки, объем файлов программы

2) стоимость, количество продаж, наличие программных продуктов аналогичного назначения

3) мобильность, надежность, эффективность, модифицируемость, коммуникативность, учет человеческого фактора

4) модифицируемость, надежность, наличие программных продуктов аналогичного назначения

8. Функциональные задачи – это

1) задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей

2) основа для разработки сервисных средств ПО (утилиты, библиотеки)

3) совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей

4) задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере

9. Алгоритм – это

1) комплекс математических вычислений для решения задачи
2) последовательность команд, предназначенных для решения задачи
3) программная реализация на компьютере решения определенной задачи

4) результат интеллектуального труда, для которого характерно творчество

10. Жизненный цикл программы – это

1) временной интервал, начиная с момента замысла программы и кончая прекращением всех видов его пользования

2) временной интервал, начиная с момента введения программы в эксплуатацию

3) промежуток времени, который определяет наиболее эффективное использование создаваемой программы

4) временная характеристика разработки программного продукта

11. Выбрать правильный ответ

a. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, необходимость обеспечения безопасности системы и данных

b. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, функции, которые должен выполнять программный продукт

c. На этапе сбора и анализа требований заказчик должен выяснить, прежде всего, сроки написания программы

d. На этапе сбора и анализа требований заказчик должен собрать литературу по разрабатываемому программному продукту

12. Предметная (прикладная) область – это

1) проблема, подлежащая решению

2) совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей

3) программная реализация решения задачи

4) точная формулировка решения задачи на компьютере с описанием входной и выходной информации

13. Задача – это

1) проблема, подлежащая решению

2) совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей

3) программная реализация решения задачи

4) точная формулировка решения задачи на компьютере с описанием входной и выходной информации

14. Техническое задание – это

1) набор правил, по которым строится ПП

2) задание, которое необходимо выполнить на ПК по техническим характеристикам

3) набор правил эксплуатации программного продукта

4) совокупность требований к программным средствам, которые могут использоваться как критерий проверки и приемки разработанного ПП

15. Программное средство - это

1) программа для удовлетворения нужд разработчиков, предназначенная для продажи

2) программа, предназначенная для многократного применения на различных объектах и разработанная любым способом

3) программная реализация решения задачи на компьютере

4) результат разработки какого-либо технического задания

16. Технологические задачи – это

1) задачи, требующие решения при реализации функций управления в рамках информационных систем предметных областей

2) основа для разработки сервисных средств ПО (утилиты, библиотеки)

3) совокупность связанных между собой функций и задач управления, с помощью которых достигается выполнение поставленных целей

4) задачи, которые ставятся и решаются при организации технологического процесса обработки информации на компьютере

17. На современном этапе выделяют 2 основных подхода к проектированию ПП. Какие?

1) структурный и процедурный

2) объектно-ориентированный и структурный

3) метод проектирования Джексона и объектно-ориентированный

4) иерархический и сетевой

18. Методами структурного проектирования являются

1) модульное программирование, нисходящее проектирование, кодирование и тестирование, структурное проектирование;

2) интегрированное и модульное проектирование;

3) функционально – ориентированное и объектно-ориентированное проектирование

4) структурное программирование, модульное проектирование, тестирование и кодирование

4) точная формулировка решения задачи на компьютере с описанием входной и выходной информации

19. В каких годах разработана основная часть документов единой системы программной документации?

- 1) 60-70-х гг
- 2) 70-80-х гг
- 3) 80-90-х гг
- 4) 60-90-х гг

20. Техническое задание – это

- 1) набор правил, по которым строится ПП
- 2) задание, которое необходимо выполнить на ПК по техническим характеристикам
- 3) набор правил эксплуатации программного продукта
- 4) совокупность требований к программным средствам, которые могут использоваться как критерий проверки и приемки разработанного ПП

21. Техническое задание, как одно из стадий разработки, состоит из трех частей.Каких?

- 1) научно-исследовательская работа, разработка эскизного проекта, разработка технического проекта
- 2) разработка программной документации, утверждение эскизного и технического проектов, подготовка и передача программы
- 3) обоснование необходимой разработки программы, научно-исследовательская работа, разработка и утверждение технического задания
- 4) разработка программной документации, утверждение выбранных методов разработки, утверждение технического проекта, кодирование

22. Одной из составных частей рабочего проекта является

- 1) разработка программ и программной документации
- 2) разработка технического задания;
- 3) выбор языка программирования

4) разработка технического проекта

23. Дайте расшифровку аббревиатуре ЕСПД.

- 1) Единая схема проектирования документов
- 2) Единая система программной документации
- 3) Единая схема программных документов
- 4) Единственная система программной документации

24. На сколько частей можно разделить ЕСПД?

- 1) 6
- 2) 8
- 3) 5
- 4) 10

25. Группа «0» ЕСПД указывает на ...

- 1) резервные группы
- 2) общее положение
- 3) основополагающие стандарты
- 4) прочие стандарты

26. Обозначение стандарта ЕСПД должно состоять из трех частей, где первое число обозначает...

- 1) присвоение к классу стандартов ЕСПД
- 2) код классификационной группы стандарта
- 3) год регистрации стандарта
- 4) номер документа в реестре

27. На современном этапе выделяют 2 основных подхода к проектированию ПП. Какие?

- 1) структурный и процедурный

- 2) объектно-ориентированный и структурный
- 3) метод проектирования Джексона и объектно-ориентированный
- 4) иерархический и сетевой

28. Методами структурного проектирования являются

- 1) модульное программирование, нисходящее проектирование, кодирование и тестирование, структурное проектирование;
- 2) интегрированное и модульное проектирование;
- 3) функционально – ориентированное и объектно-ориентированное проектирование
- 4) структурное программирование, модульное проектирование, тестирование и кодирование

29. Что не использует структурный подход проектирования программного продукта?

- 1) диаграммы декомпозиции
- 2) интегрированную структуру данных предметной области
- 3) структурные схемы
- 4) анализ предметной области

30. Объектно-ориентированный подход проектирования программного продукта основан на:

- 1) проектировании
- 2) кодировании и тестировании
- 3) создании иерархии классов, наследовании свойств объектов и методов их обработки
- 4) выделении классов объектов

31. Проектирование – это

- 1) итерационный процесс, при помощи которого требования к программным средствам транслируются в инженерное представление
- 2) процесс построения модели будущего программного средства
- 3) инженерное представление программного продукта на каком-либо языке программирования
- 4) представление программного продукта совокупностью объектов и их свойств

32. Назовите виды моделей управления.

- 1) модель потока данных и модель хранилища данных
- 2) модель клиент-сервер и модель управления прерываниями
- 3) модель централизованного и событийного управления
- 4) модель централизованного и периферийного управления

33. В основе модели потока данных лежит –

- 1) сцепление компонентов
- 2) разделение данных
- 3) разбиение по функциям
- 4) выделение отдельных компонентов и их свойств

34. Структурный метод проектирования ПП основан на:

- 1) разбиении единой системы на автономные объекты реального мира
- 2) выделении объектов и их методов
- 3) разбиении всей программы на отдельные логические части
- 4) последовательной декомпозиции всей системы на отдельные

компоненты

35. Исходными данными для структурного метода проектирования ПП являются

- 1) логические компоненты, составляющие программное средство

- 2) отдельные компоненты модели анализа программных средств
- 3) специальные компоненты, имеющие в основе собственный набор данных
- 4) объекты системы, их свойства и методы

36. Модуль — это...

- 1) самостоятельная часть программы, имеющая определенное назначение и обеспечивающая заданные функции обработки автономно от других программ
- 2) упорядоченный набор команд, обеспечивающий выполнение определенных функций
- 3) алгоритм построения программного продукта

37. Модульная программа – это программа, в которой...

- 1) модульную часть логической структуры можно изменить, не внося изменений в основную часть программы
- 2) модульную часть программы можно оставить без изменений, а внести изменения в саму программу
- 3) модульная часть функционирует независимо от программы

38. Объектный модуль – это...

- 1) отдельная программа, независима от других выполняемых программ
- 2) модуль специальной структуры, созданный при компиляции программы
- 3) обычный текстовый файл с нужным расширением
- 4) средство языка программирования служащее для увеличения уровня языка программирования

39. Загрузочный модуль – это...

- 1) отдельная программа, независима от других выполняемых программ

- 2) выполняемый *.exe модуль, фактически отдельная программа
- 3) обычный текстовый файл с нужным расширением
- 4) средство языка программирования служащее для увеличения уровня языка программирования

40. Сцепление модуля - это

- 1) мера его зависимости по данным от других модулей
- 2) набор откомпилированных программ, собранных в специально форматированный файл
- 3) практическая часть текста программы, которая подставляется при каждом вызове, увеличивая ее размер
- 4) степень взаимосвязи локальных переменным модуля и глобальных переменных основной программы

41. Управляющий модуль -

- 1) обеспечивает вызов других модулей на обработку
- 2) управляет запуском программного продукта
- 3) выполняют функции обработки
- 4) осуществляют обслуживающие функции

42. Какую связь устанавливает пользовательский интерфейс?

- 1) серверную
- 2) доменную
- 3) клиент-серверную
- 4) клиентскую

43. Каким базовым требованиям должен отвечать интерфейс, чтобы он был удобным?

- 1) понятность, эстетичность, устойчивость к неполадкам

2) простота, надежность, предсказуемость, адаптивность, стандартность

3) легкая восприимчивость пользователями, несхожесть с другими интерфейсами

4) предсказуемость, модифицируемость, адаптивность, эффективность

44. Язык программирования – это...

1) некоторая структура, в которой указаны правила синтаксиса и семантики

2) множество текстов некоторого алфавита, удовлетворяющих правилам синтаксиса и семантики

3) шаблон построения программных продуктов, с заранее указанными правилами синтаксиса и семантики

4) совокупность правил описания алгоритма

45. Алфавит языка программирования включает в себя

1) буквы и специальные знаки

2) зарезервированные слова

3) оба варианта правильны

46. Что такое консольное приложение?

1) это монитор и клавиатура, рассматриваемые как единое устройство

2) устройством вывода

3) программа, предназначенная для работы в операционной системе MS-DOS

4) обычное приложение Delphi

47. Объектно-ориентированное программирование строится на 3-х принципах:

1) на абстрагировании, инкапсуляции, модульности

- 2) на структурированности, абстрагировании, кодировании
- 3) на инкапсуляции, логике, модульности

48. Абстракция концентрирует внимание ...

- 1) на внутренние представления объекта
- 2) на внешние представления объекта
- 3) на определении способности подвергать декомпозиции

49. Что такое цикл?

- 1) Оператор выбора
- 2) Оператор условия
- 3) Оператор повторений
- 4) Оператор перехода

50. Какой из приведенных ниже циклов, является циклом с параметром?

- 1) WHILE
- 2) FOR
- 3) DO
- 4) REPEAT

51. Какой из приведенных ниже циклов, является циклом с предусловием?

- 1) CASE
- 2) FOR
- 3) REPEAT
- 4) WHILE

52. Какой из приведенных ниже циклов, является циклом с постусловием?

- 1) REPEAT

- 2) FOR
- 3) WHILE
- 4) VAR

53. Какое зарезервированное слово обозначает, что переменная именно символьного типа?

- 1) CHR
- 2) CHAR
- 3) BOOLEAN
- 4) SUCC;

54. Общий вид оператора IF?

- 1) If <условие> Then <оператор1> Else <оператор2>
- 2) If <оператор1> Then < оператор2> Else < условие >
- 3) If < условие > Then < оператор2> Else < оператор1>
- 4) If <параметры> Then <условие> Else <оператор>

55. Как выглядят комментарии в программе TP?

- 1) ()
- 2) { }
- 3) //
- 4) Begin ...End

56. Раздел VAR это ...

- 1) раздел описания меток
- 2) раздел описания переменных
- 3) раздел описания констант
- 4) раздел описания модулей

57. Раздел CONST это ...

- 1) раздел описания меток
- 2) раздел описания переменных
- 3) раздел описания констант
- 4) раздел описания модулей

58. Под ошибкой подразумевается

1) место в программе, где искажение проявляется или становится очевидным

2) неправильность, погрешность или неумышленное искажение объекта или процесса

3) место в программе, где возникают условия для появления искажений

4) исправление выявленных искажений в процессе тестирования программы

59. Источником ошибок в программе может быть

1) недостаточная квалификация специалиста

2) сложность программы

3) большой объем программы

4) недостаточное знание заказчиком предметной области

60. Контроль (проверка, испытание) программы является методом борьбы с...

1) переводом программы

2) взаимопониманием

3) сложностью программы

4) описанием программы

61. Выделяют следующие виды ошибок программ:

1) синтаксические, семантические, первичные

2) ошибки анализа, общего и физического характера

3) ошибки анализа, первичные и вторичные

4) ошибки описания, определения функций и кодирования

62. Под отладкой понимается процесс

1) нахождения и исправления ошибок

2) позволяющий получить программу, которая функционирует с требуемыми характеристиками

3) оптимизации программы

4) тиражирования программы

63. Для тестирования программы используют

1) простые тестовые данные

2) просчитанные данные

3) сложные данные

4) произвольные данные

64. Отладка бывает:

1) ручная и семантическая

2) ручная и автоматизированная

3) разрушающая и неразрушающая

4) разрушающая, семантическая, оптимизирующая

65. Тестирование – это...

1) процесс создания загрузочного файла программы

2) запуск программы на выполнение

3) процесс многократного выполнения программы с целью обнаружения максимального количества ошибок

4) процесс нахождения и исправления ошибок

66. Тестирование бывает

1) нисходящее, восходящее, промежуточное, завершённое

- 2) структурное, функциональное, промежуточное, полное
- 3) нисходящее, восходящее, структурное, полное
- 4) нисходящее, восходящее, структурное, функциональное

67. Использование одного имени для задания общих для класса действий, что означает способность объектов выбирать внутренний метод, исходя из типа данных, определяет свойство ООП

- 1) Полиморфизм
- 2) Управление событиями
- 3) Инкапсуляция
- 4) Наследование

68. Понятие «инкапсуляция» относится к

- 1) Технологии модульного программирования
- 2) Технологии объектно – ориентированного программирования
- 3) Технологии императивного программирования
- 4) Технологии модульного программирования

69. Свойство ООП, которое может быть смоделировано с помощью таксономической классификационной схемы (иерархии) называется

- 1) Инкапсуляция
- 2) Управление событиями
- 3) Полиморфизм
- 4) Наследование

70. Понятие класса в ООП включает в себя

- 1) Поля и методы класса
- 2) Процедуры и функции обработки
- 3) Поля и функции обработки
- 4) Поля и процедуры обработки

71. Понятия объекта в ООП - это

- 1) представитель класса
- 2) конкретные данные, заданные в классе.
- 3) компонент панели инструментов
- 4) встроенный объект Delphi

72. IDE – это...

- 1) История развития Delphi
- 2) Приложение, разрабатываемое программистом
- 3) Интегрированная среда разработки

73. Названия процедур на определенное событие:

- 1) Компоненты
- 2) Обработчики событий
- 3) Редактор кода

74. Назначение главного меню:

- 1) Осуществление функций управления при разработке программ
- 2) Сопровождение программных продуктов
- 3) Автоматизирует процесс тестирования программ

75. Примером декларативного языка программирования является...

- 1) Ассемблер
- 2) Java
- 3) Fortran
- 4) HTML
- 5) Lisp

76. Примером процедурного языка программирования является...

- 1) Ассемблер
- 2) Java
- 3) Fortran

4) HTML

5) Lisp

77. Примером объектно-ориентированного языка программирования является...

1) Ассемблер

2) Java

3) Fortran

4) HTML

5) Lisp

78. Не является языком программирования...

1) Ассемблер

2) Java

3) Fortran

4) HTML

5) Lisp

79. Какая из конструкций обозначает остаток от деления?

1) Div

2) Mod

3) If

4) Case

5) Char

80. Какая из конструкций обозначает целочисленное деление?

1) Div

2) Mod

3) If

4) Case

5) Char

81. Какая из конструкций обозначает тип данных?

1) Div

2) Mod

- 3) If
- 4) Case
- 5) Char

82. Метод, который не относится к варианту сортировки это ...

- 1) метод дихотомии
- 2) метод Хоара
- 3) бинарный метод
- 4) метод пузырька
- 5) метод вставки

83. Метод быстрой сортировки, который обычно связан с рекурсией это ...

- 1) метод дихотомии
- 2) метод Хоара
- 3) бинарный метод
- 4) метод пузырька
- 5) метод вставки

84. Метод быстрой сортировки, который связан с построением дерева это ...

- 1) метод дихотомии
- 2) метод Хоара
- 3) бинарный метод
- 4) метод пузырька
- 5) метод вставки

85. Какая связка операторов не является циклом?

- 1) For ... to ... do
- 2) For ... downto ... do
- 3) Repeat ... until
- 4) While ... do
- 5) with ... do

86. Какое из выражений дается в блок-схеме алгоритма внутри ромба?

- 1) $A > 2$
- 2) начало
- 3) $N = N + 1$
- 4) 1
- 5) $N = 1, 10$

87. Какое из выражений дается в блок-схеме алгоритма внутри шестиугольника?

- 1) $A > 2$
- 2) начало
- 3) $N = N + 1$
- 4) 1
- 5) $N = 1, 10$

88. Какое из выражений дается в блок-схеме алгоритма внутри прямоугольника?

- 1) $A > 2$
- 2) начало
- 3) $N = N + 1$
- 4) 1
- 5) $N = 1, 10$

89. Какое из выражений дается в блок-схеме алгоритма внутри овала?

- 1) $A > 2$
- 2) начало
- 3) $N = N + 1$
- 4) 1
- 5) $N = 1, 10$

90. Какое из выражений дается в блок-схеме алгоритма внутри круга?

- 1) $A > 2$
- 2) начало
- 3) $N = N + 1$
- 4) 1
- 5) $N = 1, 10$

91. Названием стандартной функции вычисления натурального логарифма ...

- 1) Ln
- 2) Round
- 3) Abs
- 4) Sqrt
- 5) Sqr

92. Названием стандартной функции вычисления модуля числа ...

- 1) Ln
- 2) Round
- 3) Abs
- 4) Sqrt
- 5) Sqr

93. Названием стандартной функции вычисления квадратного корня

...

- 1) Ln
- 2) Round
- 3) Abs
- 4) Sqrt
- 5) Sqr

94. Названием стандартной функции вычисления $x * x$...

- 1) Ln
- 2) Round
- 3) Abs
- 4) Sqrt

5) Sqr

95. Названием стандартной функции округления до целого числа ...

1) Ln

2) Round

3) Abs

4) Sqrt

5) Sqr

96. Какой командой подключается генератор случайных чисел?

1) Randomize

2) Graph

3) Nod

4) Dispose

97. Что такое комментарий?

1) оператор TP

2) пояснения к программе и для программы

3) операторные скобки

4) пояснения к программе для программиста

98. Что такое алгоритм?

1) Последовательность команд исполнителю, выполнение которых приводит к решению поставленной задачи.

2) Последовательность действий для ЭВМ.

3) Последовательность

4) Решение задачи

99. Способы задания алгоритма?

1) Линейный, разветвляющийся

2) Табличный, словесный, блок-схема

3) Табличный, блок-схема

4) Линейный, циклический, разветвляющийся

100. Типы алгоритма?

- 1) Табличный, блок-схема, словесный
- 2) Линеинный, циклический, разветвляющийся
- 3) Табличный, блок-схема
- 4) Линеинный, циклический,

Задания в открытой форме

- 1) Перечислите инструментальные среды программирования.
- 2) Перечислите основные характеристики среды программирования.
- 3) Какой интерфейс пользователя является объектно-ориентированным?
- 4) Дайте понятие алгоритма, свойства и классификация алгоритмов.
- 5) Какая форма диалога интерфейса предполагает использование команд специально разработанного формального языка?
- 6) Для какого интерфейса наиболее характерно использование компонентов: кнопки, строки ввода и вывода, списки, реверсивные счетчики, переключатели, флажки?
- 7) Какие условия работы операторов ветвления if и case?
- 8) Какой подход к тестированию программного обеспечения не предусматривает знания блок-схемы алгоритма?
- 9) Каким образом тип диалога интерфейса связан с формой диалога интерфейса?
- 10) Какой критерий структурного тестирования не всегда приводит к выполнению каждого оператора хотя бы 1 раз?
- 11) При каком методе функционального тестирования используется математический аппарат алгебры логики?
- 12) К какому виду ошибок относится следующая: предусмотрено обращение к подпрограмме другого модуля, но при объединении модулей не стыкуются списки параметров?

- 13) При каком методе отладки программного обеспечения вначале формируют множество причин, которые могут вызвать ошибку. Затем анализируют причины и исключают те, которые противоречат имеющимся данным.
- 14) Какой метод получения дополнительной информации об ошибке не эффективен при больших и сложных программах?
- 15) Выберите правильную последовательность этапов отладки программного обеспечения.
- 16) Какой программный документ должен содержать сведения для установки программного обеспечения, его настройки и проверки работоспособности?
- 17) В чем различие между перегрузкой методов и переопределением методов?
- 18) Какие структуры данных можно размещать как в статической, так и в динамической памяти?
- 19) Какая диаграмма отражает физические взаимосвязи между программными и аппаратными компонентами системы?
- 20) С какой диаграммы следует начинать проектирование программного обеспечения при объектном подходе?

Задание на установление соответствия

1. Установить соответствие

1) Файл проекта	a) (.pas) соответствующий файл модуль для хранения кода
2) Файл модуля	b) (.res) содержит пиктограмму и прочие ресурсы
3) Файл формы	c) (.dpr) текстовый файл используется для хранения информации о формах и модулях, содержит операторы инициализации и запуска программы на выполнение
4) Файл опций проекта	d) (.dfm) двоичный файл, который создается для хранения информации о ваших формах и фреймах

5) Файл ресурсов	e) (.cfg)хранит установки проекта
6) Файл конфигурации проекта	f) (.dof) хранит установки опций проекта

2. Установить соответствие:

1) Application.Initialize	a) загружает и инициализирует форму
2) Application.Create Form	b) активизирует форму и начинает выполнение приложения
3) Application.Run	c) подготавливает приложение к работе

3. Установить соответствие:

1) Тестирование уязвимости	a) При таком тестировании система подвергается предельным нагрузкам за счёт повышения интенсивности поступления входных данных и (или) за счёт ограничения системных ресурсов (процессорного времени, памяти, дискового пространства и т. д.).
2) Нагрузочное тестирование	b) Направлено на проверку безопасности и защищённости программных продуктов. Оно проводится с целью проверки эффективности используемых в программах механизмов защиты информации, их устойчивости к атакам, а также к ошибкам персонала.
3) Функциональное тестирование	c) Повторное тестирование всей программы после внесения изменений в её часть.
4) Регрессионное тестирование	d) Проверяются только функциональные возможности системы, то есть полнота и правильность реализации функциональных требований. Нефункциональные характеристики (надёжность, эффективность, защищённость и т. д.) целенаправленно не проверяются.

4. Установить соответствие:

1) Альфа-тестирование	а) Выполняется силами большого количества пользователей, которые не являются сотрудниками организации. За счёт массовости этого процесса выявляются ошибки, упущенные штатными тестерами.
2) Бета-тестирование	б) Обозначают всю деятельность организации-разработчика по тестированию данного продукта, которую она выполняет своими силами.
3) Тестирование черного ящика	в) При тестировании тестер осуществляет формирование теста с учетом знания об испытываемом объекте; например, тест для компьютерной программы составляется с учетом практически полного знания о тестируемом модуле (о его структуре, конфигурации, алгоритмах работы и т. д.).
4) Тестирование белого ящика	г) При тестировании о внутреннем устройстве тестируемого объекта неизвестно ничего, и тестирование осуществляется только на основе внешней спецификации.

5. Установить соответствие:

1) Дымовое тестирование	а) Выполняется силами большого количества пользователей, которые не являются сотрудниками организации. За счёт массовости этого процесса выявляются ошибки, упущенные штатными тестерами.
2) Комплексное тестирование	б) Заключается в изолированной проверке каждого отдельного элемента программной системы: компонента, модуля, подпрограммы.
3) Автономное тестирование	в) Проверяется взаимодействие частей программы и их взаимное влияние. Для упрощения процедуры рекомендуется сначала проводить тестирование по отдельным функциональным группам, а затем провести тестирование в целом.

6. Установить соответствие:

1) Ошибки анализа	a) В рамках предложенного проекта невозможно достичь требуемой в ТЗ функциональности и критериев качества.
2) Ошибки проектирования	b) Это самые тяжёлые по последствиям ошибки. Примеры: не предусмотрена нужная функциональность; дублирование функций; лишние функции; неверно оценены требования к техническим средствам; неверно оценены требования к производительности или переносимости.
3) Ошибки документации	c) Расхождения документов с программами, отсутствие нужных сведений и нужных документов
4) Ошибки программной реализации.	d) Неточности в программном коде, дизайне или реализации, допущенные в процессе разработки и написания программы.

7. Установить соответствие:

1) Ошибки пользовательского интерфейса	a) Программа не учитывает возможность ошибок, вызванных неверным порядком выполнения действий (например, принтер ещё не выбран, а печать вызывается) и т. п. Типичная ситуация – повторный вызов функции или программы во время её выполнения.
2) Ошибки вычислений	b) Каждый этап должен сопровождаться описанием или оценкой вычислительных погрешностей и других ограничений, снижающих его универсальность. Отсутствие этого может привести к неверной работе программы, реализующей данный метод.
3) Ошибки использования методов	c) Программа не учитывает возникновение или накопление ошибок (погрешности, округления, переполнения и пр.)
4) Ошибки	d) Не позволяет использовать какие-либо

синхронизации	существующие функциональные возможности (целиком или частично). Например, реализовано несколько методов, невозможно выбрать нужный.
---------------	---

8. Установить соответствие:

1) A:=3.234	a) Integer
2) Z:=14	b) Real
3) B:=10351	c) Byte
4) K:=-0.3	d) Word

9. Установить соответствие:

1) Var	a) Раздел операторов
2) Program	b) Раздел имени программы
3) Uses	c) Раздел описания переменных
4) Begin...end	d) Раздел описания библиотечных модулей

10. Установить соответствие:

1) Машинные коды	a) Программа записывается служебными словами, разными для разных моделей процессора
2) Языки программирования высокого уровня	b) Программа представляет собой последовательность нулей и единиц
3) Языки ассемблера	c) Программа не привязана к конкретным моделям оборудования

11. Установить соответствие:

1) Командный интерфейс	а) Движения
2) SILK	б) Последовательности символов
3) WIMP	с) Манипулятор

12. Установить соответствие:

1) Декларативный подход	а) Подход подразумевает написание программного кода без использования объектов.
2) Рекурсивный подход	б) Парадигма программирования, в которой основными концепциями являются понятия объектов и классов (либо, в менее известном варианте языков с прототипированием — прототипов).
3) Объектно-ориентированный подход	с) Подразумевает разделение сложной задачи, на один простой шаг к её решению и оставшуюся часть, которая становится упрощённой версией той же задачи.
4) Процедурный подход	д) Выражается логика вычисления без отсутствия описания потока управления.

13. Установить соответствие:

1) floattostr	а) Преобразовать строку в целое число
2) strtfloat	б) Конвертирует целое число в строку
3) inttostr	с) конвертирует числовую строку, в значение с плавающей точкой
4) strtoint	д) Преобразует значение с плавающей запятой в строку

14. Установить соответствие:

<p>1) «стихийное» программирование</p>	<p>а) Определяется как технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности <i>объектов</i>, каждый из которых является экземпляром определенного типа (<i>класса</i>), а классы образуют иерархию с <i>наследованием</i> свойств.</p>
<p>2) структурный подход к программированию</p>	<p>б) практически отсутствовали сформулированные технологии, и программирование фактически было искусством. Первые программы имели простейшую структуру.</p>
<p>3) объектный подход к программированию</p>	<p>с) Подход к программированию представляет собой совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки программного обеспечения.</p>

15. Установить соответствие:

<p>1) OLE-automation или просто Automation</p>	<p>а) Технология, организующая доступ к данным разных компьютеров с учетом балансировки нагрузки сети.</p>
<p>2) ActiveX</p>	<p>б) Технология, обеспечивающая безопасность и стабильную работу распределенных приложений при больших объемах передаваемых данных.</p>
<p>3) MIDAS</p>	<p>с) Технология предназначена для создания программного обеспечения как сосредоточенного на одном компьютере, так и распределенного в сети.</p>
<p>4) MTS (Microsoft Transaction Server)</p>	<p>д) Технология создания программируемых приложений, обеспечивающая программируемый</p>

	доступ к внутренним службам этих приложений
--	---

16. Установить соответствие:

1) Начальный уровень	а) Характеризуется тем, что мероприятия по улучшению применяются не только к существующим процессам, но и для оценки эффективности ввода новых технологий. Основной задачей всей организации на этом уровне является постоянное улучшение существующих процессов. При этом улучшение процессов в идеале должно помогать предупреждать возможные ошибки или дефекты.
2) Повторяемый уровень	б) Характеризуется тем, что стандартный процесс создания и сопровождения программного обеспечения полностью документирован (включая и разработку по, и управление проектами). Подразумевается, что в процессе стандартизации происходит переход на наиболее эффективные практики и технологии. Для создания и поддержания подобного стандарта в организации должна быть создана специальная группа.
3) Определенный уровень	с) На предприятии внедрены технологии управления проектами. При этом планирование и управление проектами основывается на накопленном опыте, существуют стандарты на разрабатываемое программное обеспечение (причем обеспечивается следование этим стандартам) и специальная группа обеспечения качества.
4) Оптимизирующий уровень	д) Описан в стандарте в качестве основы для сравнения со следующими уровнями. На предприятии такого уровня организации не существует стабильных условий для создания качественного программного

	<p>обеспечения. Результат любого проекта целиком и полностью зависит от личных качеств менеджера и опыта программистов, причем успех в одном проекте может быть повторен только в случае назначения тех же менеджеров и программистов на следующий проект.</p>
--	--

17. Установить соответствие:

1) Линейная структура процесса вычислений	a) Предполагает, что для получения результата некоторые действия необходимо выполнить несколько раз.
2) Разветвленная структура процесса вычислений	b) Предполагает, что конкретная последовательность операций зависит от значений одной или нескольких переменных.
3) Циклическая структура процесса вычислений	c) Предполагает, что для получения результата необходимо выполнить некоторые операции в определенной последовательности.

18. Установить соответствие:

1) правильность	a) Возможность проверки получаемых результатов;
2) универсальность	b) Обеспечение полной повторяемости результатов, т. Е. Обеспечение их правильности при наличии различного рода сбоях;
3) надежность	c) Обеспечение правильной работы при любых допустимых данных и защиты от неправильных данных;
4) проверяемость	d) Функционирование в соответствии с техническим заданием;

19. Установить соответствие:

1) Точность результатов	a) Возможность совместного функционирования с некоторым оборудованием
2) Защищенность	b) Возможность совместного функционирования с другим программным обеспечением
3) Программная совместимость	c) Обеспечение конфиденциальности информации;
4) Аппаратная совместимость	d) Обеспечение погрешности результатов не выше заданной;

20. Установить соответствие:

1) Прimitивный интерфейс	a) Реализуют множество сценариев, представленных в операциях над объектами, основные операции инициируются перемещением пиктограмм объектов мышью, данная форма реализована в интерфейсе самой операционной системы windows альтернативно интерфейсу со свободной навигацией;
2) Интерфейс меню	b) Реализуют множество сценариев, операции которых не привязаны к уровням иерархии, и предполагают определение множества возможных операций на конкретном шаге работы; интерфейсы данной формы в основном используют windows-приложения;
3) Интерфейс со свободной навигацией	c) Реализуют множество сценариев работы, операции которых организованы в иерархические структуры, например, «вставка»: «вставка файла», «вставка символа» и т. Д.;
4) Интерфейс прямого манипулирования	d) Реализуют единственный сценарий работы, например, ввод данных - обработка - вывод результатов;

Задание на установление правильной последовательности

1. Установить основные этапы развития программирования, как науки:
 1. «стихийное» программирование.
 2. Структурный подход к программированию
 3. Объектный подход к программированию
 4. Компонентный подход и case-технологии

2. Установить этапы разработки:
 1. Проектирование
 2. Реализация
 3. Внедрение
 4. Анализ и планирование требований пользователей

3. Установить этапы отладки программного обеспечения:
 1. Изучение проявления ошибки
 2. Локализация ошибки
 3. Определение причины ошибки
 4. Исправление ошибки
 5. Повторное тестирование

4. Установить этапы тестирования:
 1. Создание тестовой документации. Написание сценариев, которые позволят проверить функционал.
 2. Тестирование прототипа. Тестирование основного функционала продукта, корректировка целей, добавление фичей.
 3. Основное тестирование. Выполнение общей проверки продукта.
 4. Стабилизация. На данном этапе происходит работа над устранением багов.

5. Эксплуатация. Проводится регресс-тестирование, устранение ошибок, которые нашел конечный пользователь.
6. Работа с требованиями. Знакомство с требованиями заказчика, что должен из себя представлять итоговый продукт, обсуждение.
7. Разработка стратегии тестирования. Оценка сроков тестирования, выявление среды тестирования, объединение всей информации, полученной при работе с требованиями.

5. Установить цели и содержание этапов при разработке ПО:

1. Ручное кодирование
2. Ручное документирование
3. Тестирование кодов
4. Сопровождение кодов
5. Традиционная разработка (Основные усилия - на
6. кодирование и тестирование)
7. 'Бумажные' спецификации

6. Установить цели и содержание этапов при разработке ПО CASE-технологии:

1. Автоматический контроль проекта
2. Сопровождение спецификаций проектирования
3. Быстрое итеративное прототипирование
4. Автоматическая кодогенерация
5. Автоматическая генерация документации

7. Установить этапы отладки:

1. Комплексная динамическая отладка, в том числе и в режиме реального времени;
2. Диагностика и устранение причин отклонений результатов от эталонных значений;

3. Создание совокупности тестовых эталонных заданий и значений, которым должна соответствовать программа;
 4. Статистическая проверка текстов программы;
 5. Тестирование и выполнение программы с различными уровнями детализации;
 6. Изменение программы с целью исправления ошибок.
8. Установить этапы работы, выполняемые в процессе эволюционного прототипирования:
1. Разработка прототипа системы
 2. Разработка обобщённой спецификации
 3. Использование прототипа системы
 4. Доработка прототипа системы
 5. Проверка системы
 6. Получение законченной системы
9. Установить этапы работы, выполняемые в процессе внесения изменений в спецификацию требований:
1. Внесение изменений в код системы
 2. Разработка требования
 3. Анализ требования, его стоимость
 4. Внесение изменения в спецификацию
 5. Оформление изменения
10. Установить этапы процесса проектирования интерфейса:
1. Разработка динамического прототипа
 2. Разработка исполняемого прототипа
 3. Оценка проекта пользователями
 4. Реализация интерфейса

11. Установить этапы работы, выполняемые в процессе инспектирования:

1. Планирование
2. Индивидуальная подготовка
3. Предварительный просмотр
4. Доработка
5. Исправление ошибок

12. Установить этапы выполнения основных операций, их приоритет выполнения:

1. &&
2. %=
3. ,
4. ()

13. Установить этапы выполнения основных операций, их приоритет выполнения:

1. ->
2. --
3. *
4. .

14. Установить основные этапы проектирования:

1. Проектирование интерфейсов
2. Архитектурное проектирование
3. Обобщённая спецификация
4. Проектирование алгоритмов
5. Компонентное проектирование
6. Проектирование структур данных

15. Установить этапы работы компилятора:

1. Генерация промежуточного кода,
2. Оптимизация кода и генерация кода.
3. Лексический анализ,
4. Синтаксический анализ,
5. Семантический анализ,

16. Установить этапы жизненного цикла программного обеспечения:

1. Программирование (создание программного кода), тестирование, автономная и комплексная отладка программ.
2. Документирование программного продукта, подготовка эксплуатационной и технологической документации.
3. Выход на рынок программных средств, распространение программного продукта.
4. Эксплуатация программного продукта пользователями.
5. Сопровождение программного продукта.
6. Маркетинг рынка программных средств, спецификация требований к программному продукту.
7. Проектирование структуры программного продукта.

17. Установить порядок этапов компьютерного моделирования:

1. Анализ результатов моделирования.
2. Разработка модели.
3. Постановка задачи.
4. Компьютерный эксперимент.

18. Установить этапы разработки программной документации:

1. Разработка технического проекта.
2. Комплексное внедрение программной документации.
3. Подготовка технического специального задания.
4. Составление подробного эскизного варианта проекта.

5. Оформление рабочего документа.

19. Установить этапы компиляции:

1. ... анализ. Последовательность символов исходного файла преобразуется в последовательность лексем.
2. ... анализ. Дерево разбора обрабатывается с целью установления его семантики (смысла) — например, привязка идентификаторов к их декларациям, типам, проверка совместимости, определение типов выражений и т. д.
3. Выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла.
4. Генерация кода.
5. ... анализ. Последовательность лексем преобразуется в дерево разбора.

20. Установить этапы выполнения основных операций, их приоритет выполнения:

1. []
2. ++
3. %
4. >>

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

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

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

Общий балл по промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по 5-балльной шкале следующим образом:

Соответствие 100-балльной и 5-балльной шкал

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

2.2 КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ЗАДАЧИ

1. Заполнить массив нулями, кроме первого и последнего элементов, которые должны быть равны единице.
2. Заполнить массив нулями и единицами, при этом данные значения чередуются, начиная с нуля.
3. Сформировать строку из 15 символов. На нечетных позициях должны находиться нечетные цифры, на четных позициях - буквы.
4. Заполнить массив последовательными нечетными числами, начиная с единицы.
5. Определить, содержит ли массив данное число x
6. Найти количество четных чисел в массиве.
7. Найти количество чисел в массиве, которые делятся на 3, но не делятся на 7.
8. Дана строка. Вывести ее три раза через запятую и показать количество символов в ней.
9. Дана строка. Вывести первый, последний и средний (если он есть) символы.

10. Дана строка. Вывести первые три символа и последний три символа, если длина строки больше 5. Иначе вывести первый символ столько раз, какова длина строки.

11. Заполнить массив последовательными четными числами, начиная с пяти.

12. Сформировать строку из 10 символов. На четных позициях должны находиться четные цифры, на нечетных позициях - буквы.

13. Создать матрицу 3 x 4, заполнить ее числами 0 и 1 так, чтобы в одной строке была ровно одна единица, и вывести на экран.

14. Дана матрица. Вывести на экран вторую строку.

15. Создать и вывести на экран матрицу 2 x 3, заполненную случайными числами из [0, 9].

16. Найти количество нечетных чисел в массиве.

17. Дана матрица. Вывести на экран первый и последний столбцы.

18. Дана матрица. Вывести на экран первую и последнюю строки.

19. Дана матрица. Вывести на экран все четные строки, то есть с четными номерами.

20. Дана строка. Вывести второй, предпоследний и третий символы.

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

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

Балл, полученный обучающимся за решение компетентностно-ориентированной задачи, суммируется с баллом, выставленным ему по результатам тестирования. Общий балл промежуточной аттестации суммируется с баллами, полученными обучающимся по результатам текущего контроля успеваемости в течение семестра; сумма баллов переводится в оценку по 5-балльной шкале следующим образом:

Соответствие 100-балльной и 5-балльной шкал

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

Критерии оценивания решения компетентностно-ориентированной задачи (нижеследующие критерии оценки являются примерными и могут корректироваться):

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

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

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

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