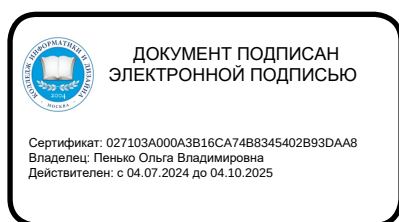




**Профессиональное образовательное частное учреждение
«КОЛЛЕДЖ ИНФОРМАТИКИ И ДИЗАЙНА»
(ПОЧУ «КИД»)**

109029, г. Москва, ул. Нижегородская, д.32, стр. 16, комн.301
ИНН 7721516041, КПП 772301001, ОГРН 1047796716990
тел. 8:(495)774-72-74, Официальный сайт kid-spo.ru

Приложение 5
к ОП СПО по специальности
09.02.07 Информационные
системы и программирование



УТВЕРЖДАЮ
Директор ПОЧУ «КИД»
О.В. Пенько
«30» августа 2024 г

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Профессионального модуля

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

программы подготовки специалистов среднего звена
по специальности

09.02.07 Информационные системы и программирование

Квалификация – Программист

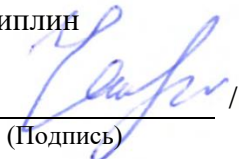
ОДОБРЕНА

Разработан на основе Федерального государственного
образовательного стандарта среднего
профессионального образования
09.02.07 Информационные
системы и программирование

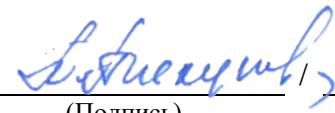
Предметно-цикловой комиссией_
общих гуманитарных и социально-
экономических, математических и общих
естественнонаучных профессиональных
дисциплин

Протокол № 1
от « 30 » августа 2024 _____

Председатель предметно-цикловой комиссии
общих гуманитарных и социально-
экономических, математических и общих
естественнонаучных профессиональных
дисциплин

 / Н. В. Чёрная
(Подпись) (Ф.И.О.)

Заместитель директора по учебно-
методической работе

 / П.В. Пискунова
(Подпись) (Ф.И.О.)

Организация-разработчик: Профессиональное образовательное частное учреждение
«КОЛЛЕДЖ ИНФОРМАТИКИ И ДИЗАЙНА»

Оглавление

<u>Пояснительная записка</u>	11
<u>1</u> <u>ФОНД ОЦЕНОЧНЫХ СРЕДСТВ МЕЖДИСЦИПЛИНАРНОГО(ЫХ) КУРСА(ОВ)</u> .	
<u>1.1</u> <u>Паспорт</u>	.
<u>1.2</u> <u>Контрольно-оценочные средства текущего контроля</u>	14
<u>1.3</u> <u>Контрольно-оценочные средства промежуточной аттестации</u>	26
<u>2</u> <u>ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ЭКЗАМЕНА по профессиональному модулю</u>	

Пояснительная записка

Фонд оценочных средств (далее – ФОС) является неотъемлемой частью нормативно-методического обеспечения системы оценки качества освоения обучающимися основной профессиональной образовательной программы подготовки специалистов среднего звена и обеспечивает повышение качества образовательного процесса.

Целью создания ФОС является установление соответствия уровня подготовки обучающихся на конкретном этапе обучения требованиями Федерального государственного стандарта среднего профессионального образования и иным нормативным документам.

ФОС по профессиональному модулю, практике представляет собой совокупность контролирующих материалов, позволяющих оценить приобретенные компетенции при проведении текущего контроля успеваемости, промежуточной аттестации обучающихся по междисциплинарному курсу, видам практики, экзамену по профессиональному модулю.

Код	Наименование компетенции
ОК 01	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 02	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
ОК 03	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
ОК 04	Эффективно взаимодействовать и работать в коллективе и команде
ОК 05	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 05	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
ОК 06	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 07	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
ОК 08	Пользоваться профессиональной документацией на государственном и иностранном языках.
ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4	Выполнять тестирование программных модулей.
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода.
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств.
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет

	соответствия стандартам кодирования
ПК 3.1	Осуществлять ревьюирование программного кода в соответствии с технической документацией.
ПК 3.2	Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям
ПК 3.3	Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма.
ПК 3.4	Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определенным техническим заданием.
ПК 4.1	Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем.
ПК 4.2	Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем.
ПК 4.3	Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика.
ПК 4.4	Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.
ПК 11.1	Осуществлять сбор, обработку и анализ информации для проектирования баз данных.
ПК 11.2	Проектировать базу данных на основе анализа предметной области.
ПК 11.3	Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.
ПК 11.4	Реализовывать базу данных в конкретной системе управления базами данных.
ПК 11.5	Администрировать базы данных
ПК 11.6	Защищать информацию в базе данных с использованием технологии защиты информации.

В результате аттестации по МДК осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих и профессиональных компетенций:

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь:	
Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля
Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля
Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования
Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства

Контроль и оценка освоения МДК по темам

Элемент МДК	Осваиваемые компетенции	
Раздел 1. Разработка программных модулей		

Тема 1.1. Жизненный цикл ПО		
Тема 1.2 Структурное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	1. Задания с выбором одного варианта правильного ответа 2. Задания с выбором нескольких вариантов правильных ответов 3. Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность 4. Задания на соответствия понятий и содержания, задания на дополнения, задания на сопоставление, задания на установление правильной последовательности
Тема 1.3 Объектно-ориентированное программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	
Тема 1.4 Паттерны программирования	ПК 1.1 ПК 1.2 ОК 1-ОК 9	
Тема 1.5 Событийно управляемое программирование	ПК 1.1 ПК 1.2 ОК 1-ОК 9	
Тема 1.6 Оптимизация и рефакторинг кода	ПК 1.1 ПК 1.2 ОК 1-ОК 9	
Тема 1.7 Разработка пользовательского интерфейса	ПК 1.1 ПК 1.2 ОК 1-ОК 9	
Тема 1.8 Основы ADO.Net	ПК 1.1 ПК 1.2 ОК 1-ОК 9	

1.1 Контрольно-оценочные средства текущего контроля

МДК.01.01 Разработка программных модулей

Раздел 1. Разработка программных модулей

Тема 1. Жизненный цикл ПО

1.Вопросы с одним вариантом правильного ответа

1.С какого момента появляется жизненный цикл?

- а) С момента осознания появления, данного ПО; +
- б) С момента его полного выхода из употребления.

2.Стадии жизненного цикла?

- а) Определения требований и спецификаций; проектирования; программирования и сопровождения; +
- б)Надежность; технологичность; правильность.

3.Сколько стадий в ЖЦ программного обеспечения?

- а) 4;
- б)5;
- в) 6. +

4.Что делает программный–аналитик?

- а) анализирует и проектирует комплекс взаимосвязанных программ; +
- б) разработчик формальных постановок задач, требующих реализации на ЭВМ.

5.Кто является разработчиком формальных постановок задач, требующих реализацию на ЭВМ?

- а) Программист–аналитик;
- б) Инженер;
- в) Постановщик задач. +

6.Что определяют функциональные спецификации?

- а) функции, которые должно выполнять ПО, т.е. в них определяется, что надо сделать системе, а не то, как это делать; +
- б)Точность не позволяет различных толкований.

7.Что такое цель этапа?

- а) иерархическое разбиение сложных задач создания ПО на подзадачи меньшей сложности; +
- б) Результатом работы на этом этапе являются спецификации на отдельные модули, дальнейшая декомпозиция которых нецелесообразна.

8.Для чего предназначен маркетинг?

- а) Результатом работы на этом этапе являются спецификации на отдельные модули, дальнейшая декомпозиция которых нецелесообразна; +
- б)Изучение также существующие аналоги и продукты- конкуренты.

9. Чем характеризуется каскадная модель?

- а) характеризуется последовательным выполнением входящих в ее состав этапов, окончанием каждого предыдущего этапа до начала последующего, отсутствием временного перекрытия этапов (последующий этап не начнется, пока не завершится предыдущий), отсутствием (или определенным ограничением) возврата к предыдущим этапам. Выявление и устранение ошибок в каскадной модели производится только на этапе тестирования; +
- б) Основной ее особенностью является наличие обратных связей между этапами, проведением проверок и корректировок проектируемого программного обеспечения на каждой стадии разработки

10. Кто такой администратор сети?

- а) человек, который обеспечивает организационную поддержку работы локальной сети; +
- б) Основной потребитель программ.

11. Основным потребителем программ является?

- а) конечный пользователь, который, как правило, не является специалистом в области программирования; +
- б) Он не программирует, а используют готовые программы для обеспечения работы на ЭВМ.

12. Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...

- а) Каскадной модели жизненного цикла ПО;
- б) Итерационной модели жизненного цикла ПО; +
- в) Спиральной модели жизненного цикла ПО;
- г) Модели жизненного цикла UML.

13. Выявлением и устранением ошибок только на стадии тестирования, которая может растянуться во времени или вообще никогда не завершиться, характеризуется ...

- а) каскадной модели жизненного цикла +
- б) итерационной модели жизненного цикла
- в) спиральной модели жизненного цикла ПО
- г) модели жизненного цикла UML

2. Вопросы с несколькими вариантами правильного ответа

1. Какие требования устанавливаются на первой стадии?

- а) по надежности; +
- б) технологичности; +
- в) правильности; +
- г) универсальности; +
- в) эффективности; +
- д) информационной согласованности; +

2. Перечислите значения спецификаций.

- а) Спецификации являются заданием на разработку ПО и их
- б) выполнение - закон для разработчика. +
- в) Спецификации используются для проверки готовности ПО. +
- г) Спецификации являются неотъемлемой частью программной документации, облегчают сопровождение и модификацию ПО. +

3. Что происходит на второй стадии?

- а) Формируется структура ПО и разрабатываются алгоритмы, задаваемые спецификациями; +
- б) Устанавливается состав модулей с разделением их на иерархические уровни на основе изучения схем алгоритмов; +
- в) Выбирается структура информационных массивов; +
- г) Фиксируются межмодульные интерфейсы; +

Тема 2. Структурное программирование

1.

Вопросы с одним вариантом правильного ответа

1. Алгоритм вычисления значений функций $F(w)$ и $Q(w)$, где w — натуральное число, задан следующими соотношениями:

- а) $F(1) = 1; Q(1) = 1;$
- б) $F(w) = F(w-1) + 2 * Q(w-1)$ при $w > 1$
- в) $Q(w) = Q(w-1) - 2 * F(w-1)$ при $w > 1.$

Чему равно значение функции $F(5) + Q(5)$?

- а) 62
- б) -14 +
- с) -12
- д) 31

2. Дан рекурсивный алгоритм:

```
a) procedure (n:integer);  
begin writeln('*');  
if n > 0 then begin  
F(n-2);  
F(n div 2);  
F(n div 2);  
end  
end;
```

Сколько символов «звездочка» будет напечатано на экране при выполнении вызова $F(5)$?

- а) 12
- б) 23
- с) 34 +
- д) 45

3. Основная идея структурного программирования состоит в том, что

- а) При написании программ не используются подпрограммы;

- b) Используется инкапсуляция и наследование объектов;
- c) Для реализации алгоритма рекомендуется использовать операторы безусловного перехода;
- d) Для написания программ используются только три типа операторов: линейный, ветвление, повторение цикла +

2. Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность.

(Внимательно прочитайте текст задания, продумайте логику и полноту ответа. Запишите ответ, в случае расчетной задачи, запишите решение и ответ)

4. Дан рекурсивный алгоритм:

Найдите сумму чисел, которые будут выведены при вызове F(2).

```
procedure F(n: integer);
begin
  writeln(n);
  if n < 5 then begin
    F(n+1);
    F(n+2);
    F(n*3)
  end
end;
```

Ответ: 79

5. Дан рекурсивный алгоритм:

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(5)?

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + F(n - 2)
  else
    F := n;
  end;
```

Ответ: 8

6. Ниже записаны две рекурсивные функции F и G:

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

```
function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + G(n - 2)
  else
    F := n;
  end;
function G(n: integer): integer;
begin
  if n > 2 then
    G := G(n - 1) + F(n - 2)
```

```
else  
G := n+1;  
end;
```

Ответ: 17

Задания и контрольные вопросы по проверке результатов практических работ

1. Описание

Проверка полученных на уроках знаний и результат формируемых умений по **Теме 2. Структурное программирование** осуществляется на практических работах. Результаты выполнения проверяется преподавателем, и оценка выставляется после ответов на контрольные вопросы практической работы).

Тема 3. Объектно-ориентированное программирование

1. Вопросы с одним вариантом правильного ответа

1. В основе концепции объектно-ориентированного программирования лежит понятие:

- a) Объекта +
- b) Класса
- c) Инкапсуляции

2. Инкапсуляция – это:

- a) Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.
- b) Сущность в адресном пространстве вычислительной системы, появляющаяся при создании экземпляра класса или копирования прототипа (например, после запуска результатов компиляции и связывания исходного кода на выполнение)
- c) Свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, и скрыть детали реализации от пользователя. +

3. Способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые.

- a) Полиморфизм
- b) Абстрагирование +
- c) Прототип

4. Термин "наследование" обозначает, что...

- a) В производных классах присутствует часть состояния родительского класса
- b) Производные классы содержат поля и методы родительского +
- c) Производные классы наследуют модификаторы доступа членов родительского класса

5. Способ защититься от использования объектов одного класса вместо другого, или по крайней мере управлять таким использованием – это:

- а) Типизация +
- б) Наследование
- с) Полиморфизм

6. В каких случаях вызывается деструктор:

- а) Создание объекта
- б) Удаление объекта +
- с) Редактирование объекта

2. Вопросы с несколькими вариантами правильных ответов

7. В каких отношениях может находиться один класс с другим:

- а) Отношение наследования +
- б) Отношение включения +
- с) Отношение использования+

3. Задания

На соответствия понятий и содержания, задания на дополнения, задания на сопоставление, задания на установление правильной последовательности (Установите соответствие, добавив в свободные поля к цифрам буквы.)

8. Соотнесите понятия

A) Состояние объекта	1. некоторый объект или множество объектов
B) Поведение объекта	2. набор методов (программный код), оперирующих над состоянием объекта
C) Значение атрибута объекта	3. набор значений его атрибутов

A-3

B-2

C-1

9. Соотнести понятия спецификаторов:

Спецификаторы	Определения
A) private	1. защищенный, разрешено обращаться как с текущего класса, так и с классов наследников
B) protected	2. закрытый, то есть к нему можно обращаться только в текущем классе
C) public	3. общедоступный, разрешено обращаться из любого места программы

A-2

B-2

C-3

10. Соотнесите определения

A) Caption	1. Определяет общий вид окна и операции с ним, которые разрешается выполнять пользователю
------------	---

Б) Icon	2. Определяет заголовок окна приложения
Б) Icon	3. Определяет значок, который будет использоваться в строке заголовка формы при выполнении программы

А-2

В-3

С-1

Задания и контрольные вопросы по проверке результатов практических работ

1. Описание

Проверка полученных на уроках знаний и результат формируемых умений по **Тема 3. Объектно-ориентированное программирование** осуществляется на практических работах. Результаты выполнения проверяется преподавателем, и оценка выставляется после ответов на контрольные вопросы практической работы).

Тема 4. Паттерны проектирования

1. Вопросы с одним вариантов правильного ответа

1. К какому из видов относиться шаблон Итератор?

- а) Порождающий
- б) Структурный
- в) Поведенческий +
- г) Базовый

2. Гарантирует, что класс имеет только один экземпляр и представляет глобальную точку доступа к нему. О каком паттерне идет речь?

- а) Фасад
- б) Одиночка+
- в) Фабричный метод
- г) Приспособленец

3. Разделяет создание сложного объекта и инициализацию его состояния так, что одинаковый процесс построения может создать объекты с разным состоянием. О каком паттерне идет речь?

- а) Строитель +
- б) Мост
- в) Адаптер
- г) Компоновщик

4. Благодаря совместному использованию, поддерживает эффективную работу с большим количеством объектов. О каком паттерне идет речь?

- а) Приспособленец +
- б) Одиночка
- в) Фасад
- г) Фабричный метод

5. Выберите верное утверждение

- а) Декоратор относится к порождающим паттернам
- б) Компоновщик относится к порождающим паттернам

- c) Мост относится к порождающим паттернам
 - d) Строитель относится к порождающим паттернам +
- 6. Предоставляет интерфейс для создания групп связанных или зависимых объектов, не указывая их конкретный класс. О каком паттерне идет речь?**
- a) Стратегия
 - b) Декоратор
 - c) Абстрактная фабрика +
 - d) Состояние

7. Что такое зависимость?

- a) описывает связь "часть"- "целое", в которой "часть" может существовать от "целого"
- b) Подвид агрегации, в которой "части" не могут существовать отдельно от "целого"
- c) Изменение в одной сущности может влиять на состояние или поведение другой сущности +
- d) Отношение наследования или реализации интерфейса

Тема 5. Событийно -управляемое программирование

1. Вопросы с одним вариантом правильного ответа

1. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки ...

- a. модель использования
- b. модель реализации +
- c. модель процессов
- d. модель развертывания

2. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения ...

- a. модель использования
- b. модель реализации
- c. модель процессов
- d. модель развертывания +

3. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Показывает, каким образом программные компоненты размещаются на конкретном оборудовании ...

- a. модель использования
- b. модель реализации +
- c. модель процессов
- d. модель развертывания

4. Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...

- a. нотация
- b. прецедент
- c. поток событий +
- d. сценарий

Тема 6. Оптимизация и рефакторинг кода

Тема 7. Разработка пользовательского интерфейса.

1. Определите, о каком принципе пользовательского интерфейса идет речь:

«Пользователь должен получать сообщения о действиях системы и о важных событиях внутри нее. Сообщения должны быть информативными, краткими, однозначными и написанными на языке, понятном пользователю.»

- a) Принцип простоты
- b) Принцип структуризации
- c) Принцип обратной связи +
- d) Принцип толерантности

2. Определите, о каком принципе пользовательского интерфейса идет речь:

«Все функции и данные, необходимые для решения определенной задачи, должны быть видны, когда пользователь пытается ее решить.»

- a) Принцип повторного использования
- b) Принцип видимости +
- c) Принцип обратной связи
- d) Принцип простоты

3. Выберите описание правила эффективности:

a) «Система должна способствовать непрерывному росту знаний, умений и навыков пользователя и приспосабливаться к его меняющемуся опыту. Плохие результаты приносит предоставление только базовых возможностей или оставление начинающего пользователя наедине со сложным интерфейсом, которым уверенно пользуются эксперты.»

b) «Система не должна препятствовать эффективной работе опытных пользователей, работающих с ней долгое время. Очевидным примером нарушения этого правила является нацеленность системы только на новичков, использование средств, которые хорошо подходят для неопытного пользователя, ограничивая его в возможности сделать что-то не так, но неэффективны для эксперта, который и так знает, что и где ему нужно сделать.» +

c) «Система должна быть настолько понятной, чтобы пользователь, никогда раньше не видевший ее, но хорошо разбирающийся в предметной области, мог без всякого обучения начать ее использовать. Это правило служит некоторым идеалом, к которому надо стремиться, поскольку на практике достичь такой степени доступности почти никогда не удастся. Тем не менее, специалисты продолжают делать в этом направлении все возможное.»

2. Вопросы с несколькими вариантами правильных ответов

4. Отметьте только принципы использования цвета, которыми нужно руководствоваться при проектировании эргономичного интерфейса:

- a) для неактивных элементов лучше использовать бледные цвета; +
- b) выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем
- c) если надо разделить данные, то цвета выбирают из различных частей спектра: красный — зеленый, синий — желтый, любой цвет — белый +

д) структура меню должна соответствовать структуре решаемой системой задачи; организация меню должна отразить наиболее эффективную последовательность шагов, ведущих к решению поставленной задачи

е) при отображении состояния, как правило, красный означает опасность (стоп), зеленый — продолжение работы, желтый — цвет предупреждения +

5. Отметьте только принципы, которыми необходимо руководствоваться при создании текстовых диалогов:

а) для неактивных элементов лучше использовать бледные цвета;

б) выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем +

с) текст в нижнем регистре читается приблизительно на 13% быстрее, чем текст, напечатанный полностью в верхнем регистре +

д) пункты меню должны быть краткими, грамматически правильными и соответствовать своему заголовку. Порядок пунктов меню выбирается в соответствии с соглашением, частотой и порядком использования, а также в зависимости от потребностей задачи или пользователя

е) для группировки данных, объединения и подбора нужно использовать соседние цвета спектра: оранжевые—желтые, синие—фиолетовые

3. Задания

На соответствия понятий и содержания, задания на дополнения, задания на сопоставление, задания на установление правильной последовательности (Установите соответствие, добавив в свободные поля к цифрам буквы.)

6. Установите соответствие

А) размер (текста, символов)	1) эффективный метод для привлечения внимания к какой-либо части экрана
В) оттенение (различная текстура объектов)	2) обычно применяют увеличение выделенного объекта в 1,5 раза
С) окружение (подчеркивание, рамки, инвертированное изображение)	3) Очень эффективный способ, если не переусердствовать

A-2

B-2

C-3

7. Установите соответствие

а) форма (символ, шрифт, форма символа)	1) эффективный метод для привлечения внимания к какой-либо части экрана
б) размер (текста, символов)	2) используется для того, чтобы дифференцировать различные категории данных
с) оттенение (различная текстура объектов)	3) обычно применяют увеличение выделенного объекта в 1,5 раза

A-2

B-3

C-1

Тема 8. Основы ADO.Net

1. Метод WriteXml для набора данных записывает данные набора данных в XML-файл. WriteXml принимает параметр для различных типов назначений. Какой из перечисленных ниже типов является недопустимым для пункта назначения?

- A. Строка
- B. System.IO.Stream
- C. System.IO.TextStream +
- D. System.IO.TextWriter

2. ADO.NET известен отключенным извлечением данных. Какой из перечисленных ниже механизмов не использует отключенный механизм при извлечении данных из базы данных?

- A. адаптерданных
- B. DataReader +
- C. Набор данных
- D. Ничего из вышеперечисленного

3. Что из перечисленного не входило в число основных целей дизайна ADO NET?

- A. Обеспечить бесперебойную поддержку XML
- B. Для прямой поддержки <url> +
- C. Обеспечить расширяемую и масштабируемую архитектуру доступа к данным для революционной многоуровневой модели программирования
- D. Расширить текущие возможности ADO

4. Наиболее распространенными в практике являются:

- A. распределенные базы данных;
- B. иерархические базы данных;
- C. сетевые базы данных;
- D. реляционные базы данных. +

5. Без каких объектов не может существовать база данных:

- A. без модулей;
- B. без отчетов;
- C. без таблиц; +
- D. без форм;
- E. без макросов;

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

6. Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какое из указанных имён файлов удовлетворяет маске:

??pri*.?*

- 1) napri.q

- 2) pripri.txt
- 3) privet.doc
- 4) 3priveta.c

Решение.

Символ «?» означает ровно один произвольный символ, значит, до «pri» должно быть не более двух символов. Этому условию удовлетворяет лишь один ответ: 1.

7. Сколько записей удовлетворяют условию «Пол = 'ж' ИЛИ Физика = 79»?

Фамилия	Пол	Математика	История	Физика	Химия	Биология
Андреев	м	80	72	68	66	70
Борисов	м	75	88	69	61	69
Васильева	ж	85	77	73	79	74
Дмитриев	м	77	85	81	81	80
Егорова	ж	88	75	79	85	75
Захарова	ж	72	80	66	70	70

Решение.

Логическое «ИЛИ» верно тогда, когда истинно хотя бы одно из утверждений. Пол = 'ж' верно для Васильевой, Егоровой и Захаровой, Физика = '79' верно для Егоровой. Логическая операция «ИЛИ» возвращает 0 только если оба аргумента имеют нулевое значение, следовательно, правильный ответ — 3.

Ответ: 3.

Во фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите ID дяди Керзона П. А. (дядя — это родной брат матери или отца).

Таблица 1		
ID	Фамилия_И.О	Пол
3272	Базилевич А. Б.	Ж
3228	Базилевич Б. Ф	М
3299	Базилевич И. Б.	М
3278	Базилевич П.И	М
3256	Базилевич Т. И.	Ж
3265	Керзон А. И.	Ж
3331	Керзон А. П.	М
3261	Керзон Л. А.	Ж
1217	Керзон П. А.	М
1202	Зельдович М. А.	Ж
3227	Витюк Д. А.	Ж
3240	Лемешко В. А.	Ж
3246	Месяц К. Г.	М
3387	Лукина Р. Г.	Ж
3293	Фокус П. А.	Ж
3332	Друк Г. Р.	Ж

Таблица 2	
ID_Родителя	ID_ребенка
3272	3272
3228	3299
3299	3272
3278	3299
3256	3240
3265	1202
3331	1217
3261	3356
1217	3378
1202	3356
3227	3378
3240	3240
3246	1202
3387	1217
3293	3261
3332	3293

Решение:

Из первой таблицы видно, что ID Керзона П. А. равен 1217. Найдем во второй таблице в графе «ID_ребенка» номер Керзона П. А. Видно, что его родители имеют ID 3272

и 3331. Родители обладателей этих ID имеют ID 3227 и 3228. Обладатели ID 3227 и 3228 имеют детей, ID которых 3299 и 3227.

Ответ: Значению ID 3299 соответствует Базилевич И. Б.

1.1 Контрольно-оценочные средства промежуточной аттестации

Перечень тем

Тема 1.1. Жизненный цикл ПО;

Тема 1.2 Структурное программирование;

Тема 1.3 Объектно-ориентированное программирование.

Тема 1.4 Паттерны программирования.

Тема 1.5 Событийно-управляемое программирование.

Тема 1.6 Оптимизация и рефакторинг кода.

Тема 1.7 Разработка пользовательского интерфейса.

Тема 1.8 Основы ADO.Net

1. Независимость программных продуктов от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п., означает их ...

- a. модифицируемость +
- b. надежность
- c. мобильность
- d. эффективность

2. Бессбойность и устойчивость в работе программных продуктов, точность выполнения предписанных функций обработки, возможность диагностики возникающих в процессе работы программ ошибок, означает их ...

- a. модифицируемость
- b. надежность +
- c. мобильность
- d. эффективность

3. Возможность применения алгоритма к целому классу однотипных задач, различающихся конкретным значением, является одним из основных свойств алгоритма ...

- a. детерминированностью
- b. дискретностью
- c. определенностью
- d. массовостью +

4. Максимально возможная интеграция программных продуктов с другими программами, обеспечение обмена данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.) означает их...

- a. модифицируемость
- b. эффективность
- c. мобильность
- d. коммуникативность +

5. Способность программных продуктов к внесению изменений, например, расширение функций обработки, переход на другую техническую базу обработки и т.п., означает их...

- a. модифицируемость
- b. эффективность
- c. мобильность +

d. учет человеческого фактора

6. Наличие дружественного интерфейса, контекстно-зависимой подсказки, хорошей документации является таким показателем качества программного продукта, как ...

- a. коммуникативность
- b. эффективность
- c. мобильность
- d. учет человеческого фактора +

7. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- постановки задачи,
- выбора критериев эффективности,
- проведения предварительных научно-исследовательских работ,
- разработки технического задания, определяет содержание этапа модели ЖЦ

- a. Эскизный проект +
- b. Технический проект
- c. Техническое задание
- d. Рабочий проект

8. В соответствии с ГОСТ 19.102-77 легальное описание действий в виде:

- структуры входных и выходных данных,
- уточнения методов решения,
- общего алгоритма,
- разработки документации эскизного проекта, определяет содержание этапа модели

ЖЦ ...

- a. Эскизный проект
- b. Технический проект +
- c. Техническое задание
- d. Рабочий проект

9. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- уточнения структуры входных и выходных данных,
- разработки алгоритмов,
- форм данных,
- семантики и синтаксиса языка,
- структуры программы,
- конфигурации технических средств,
- плана работ, определяет содержание этапа модели ЖЦ ...

- a. Эскизный проект
- b. Технический проект
- c. Техническое задание
- d. Рабочий проект +

10. В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- программирования и отладки;
- разработки документов;
- подготовки и проведения испытаний;
- корректировки программы и документов по итогам испытаний, определяет содержание этапа модели ЖЦ...

- a. Эскизный проект
- b. Технический проект
- c. Техническое задание
- d. Рабочий проект +

11. Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...

- a. каскадной модели жизненного цикла

- b. итерационной модели жизненного цикла +
- c. спиральной модели жизненного цикла ПО
- d. модели жизненного цикла UML

12. Основными особенностями:

- последовательным выполнением входящих в состав ЖЦ этапов,
- окончанием каждого предыдущего этапа до начала последующего,
- отсутствием временного перекрытия этапов,
- отсутствием возврата к предыдущим этапам, - наличием результата только в конце разработки, характеризуется...

- a. каскадной модели жизненного цикла +
- b. итерационной модели жизненного цикла
- c. спиральной модели жизненного цикла ПО
- d. модели жизненного цикла UML

13. Выявлением и устранением ошибок только на стадии тестирования, которая может растянуться во времени или вообще никогда не завершиться, характеризуется ...

- a. каскадной модели жизненного цикла +
- b. итерационной модели жизненного цикла
- c. спиральной модели жизненного цикла ПО
- d. модели жизненного цикла UML

14. Функционирование программного продукта в соответствии с техническим заданием соответствует характеристике эксплуатационных требований ...

- a. правильность
- b. универсальность
- c. надежность +
- d. адаптируемость

15. Обеспечение правильной работы при любых допустимых данных и защиты от неправильных данных соответствует характеристике эксплуатационных требований ...

- a. правильность
- b. универсальность
- c. надежность +
- d. защищенность

16. Обеспечение правильности результатов при наличии различного рода сбоев соответствует характеристике эксплуатационных требований ...

- a. правильность
- b. универсальность
- c. надежность
- d. защищенность +

17. Обеспечение погрешности результатов не выше заданной соответствует характеристике эксплуатационных требований...

- a. правильность
- b. проверяемость
- c. точность результатов +
- d. защищенность

18. Использование минимально возможного количества ресурсов технических средств (например, времени микропроцессора, объема оперативной памяти, объема внешней памяти, количества внешних устройств и др.) соответствует характеристике эксплуатационных требований...

- a. универсальность
- b. надежность
- c. аппаратная совместимость
- d. эффективность +

19. Возможность быстрой модификации с целью приспособления к изменяющимся условиям функционирования соответствует характеристике эксплуатационных требований ...

- a. универсальность
- b. эффективность
- c. адаптируемость +
- d. реентерабельность

20. Возможность «параллельного» использования несколькими процессами соответствует характеристике эксплуатационных требований ...

- a. универсальность
- b. программная совместимость
- c. повторная входимость +
- d. реентерабельность

21. Функциональная диаграмма имеет аббревиатуру ...

- a. ERD
- b. SADT +
- c. DFD
- d. STD

22. Диаграмма потоков данных имеет аббревиатуру ...

- a. ERD
- b. SADT
- c. DFD +
- d. STD

23. Диаграмма переходов состояний имеет аббревиатуру ...

- a. ERD
- b. SADT
- c. DFD
- d. STD +

24. Диаграмма «сущность–связь» имеет аббревиатуру ...

- a. ERD +
- b. SADT
- c. DFD
- d. STD

25. Вершиной древовидной структуры диаграмм, представляющей собой самое общее описание системы и ее взаимодействия с внешней средой, является следующий вид диаграммы IDEF0 ...

- a. диаграмма декомпозиции
- b. диаграмма дерева узлов
- c. контекстная диаграмма +
- d. диаграмма для экспозиции

26. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Содержит описание функций программного обеспечения с точки зрения пользователя ...

- a. модель использования +
- b. логическая модель
- c. модель реализации
- d. модель развертывания

27. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Описывает ключевые понятия моделируемого программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие его функциональность ...

- a. логическая модель

- b. модель реализации
- c. модель процессов +
- d. модель развертывания

28. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки ...

- a. модель использования
- b. модель реализации +
- c. модель процессов
- d. модель развертывания

29. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения

- a. модель использования
- b. модель реализации
- c. модель процессов
- d. модель развертывания +

30. Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования UML объединяет несколько моделей. Показывает, каким образом программные компоненты размещаются на конкретном оборудовании ...

- a. модель использования
- b. модель реализации +
- c. модель процессов
- d. модель развертывания

31. Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...

- a. нотация
- b. прецедент
- c. поток событий +
- d. сценарий

32. Подробное процедурное описание вариантов использования системы всеми заинтересованными лицами, а также внешними системами, реализует ...

- a. нотация
- b. прецедент
- c. поток событий
- d. сценарий +

33. При обращении к функции в соответствии с требованиями хорошего стиля программирования рекомендуется (но не обязательно), чтобы ...

- a. имена формальных и фактических параметров были различными
- b. количество формальных параметров соответствовало количеству фактических параметров
- c. типы формальных параметров соответствовали типам фактических параметров
- d. порядок перечисления фактических параметров соответствовал порядку перечисления формальных параметров +

34. В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...

- a. Формы
- b. Инспектора объектов +
- c. Кода
- d. Деревя объектов

35. В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...

- a. Caption
- b. Text
- c. Align
- d. Name +

36. Абстрактным называется класс, который ...

- a. не содержит функций +
- b. не содержит переменных
- c. содержит хотя бы одну виртуальную функцию
- d. содержит хотя бы одну чисто виртуальную функцию

37. Модульное тестирование предполагает ...

- a. тестирование минимально возможного для тестирования компонента, например, отдельного класса или функции
- b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы +
- c. тестирование интегрированной системы на ее соответствие исходным требованиям
- d. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

38. Интеграционное тестирование предполагает ...

- a. тестирование минимально возможного для тестирования компонента, например, отдельного класса или функции
- b. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- c. тестирование интегрированной системы на ее соответствие исходным требованиям +
- d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

39. Системное тестирование предполагает ...

- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b. тестирование интегрированной системы на ее соответствие исходным требованиям
- c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика +
- d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

40. Альфа-тестирование предполагает ...

- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b. тестирование интегрированной системы на ее соответствие исходным требованиям
- c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика +
- d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

41. Бета-тестирование предполагает ...

- a. поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы
- b. тестирование интегрированной системы на ее соответствие исходным требованиям
- c. имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика +
- d. распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

42. Каждая команда программы должна быть выполнена хотя бы один раз в соответствии с методом тестирования ...

- a. покрытие операторов;
- b. покрытие решений;
- c. покрытие условий;
- d. покрытие решений/условий; +

43. Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования ...

- a. покрытие операторов;
- b. покрытие решений;
- c. покрытие условий; +
- d. покрытие решений/условий;

44. Все возможные результаты каждого условия в решении должны быть выполнены, по крайней мере, один раз в соответствии с методом тестирования ...

- a. покрытие операторов;
- b. покрытие решений; +
- c. покрытие условий;
- d. покрытие решений/условий;

45. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...

- a. private +
- b. protected
- c. public
- d. published

46. Видимость переменных и методов класса в классах потомков определяет спецификатора доступа ...

- a. private
- b. protected +
- c. public
- d. published

47. Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...

- a. private
- b. protected
- c. public +
- d. published

48. Процесс, в ходе которого один объект может приобретать свойства другого, называется...

- a. наследование +
- b. инкапсуляция
- c. полиморфизм
- d. динамический полиморфизм

49. Процесс перегрузки функций и операторов реализует принцип ...

- a. наследования
- b. инкапсуляции
- c. статического полиморфизма +
- d. динамического полиморфизма

50. На основе наследования и виртуальных функций реализуется принцип ...

- a. наследования
- b. инкапсуляции
- c. статического полиморфизма
- d. динамического полиморфизма +

Задания с несколькими вариантами правильных ответов

1. Какие требования устанавливаются на первой стадии?

- a) По надежности; +
- b) Технологичности; +
- c) Правильности; +
- d) Универсальности; +
- e) Эффективности; +
- f) Информационной согласованности; +

2. Перечислите значения спецификаций.

- a) Спецификации являются заданием на разработку ПО и их выполнение - закон для разработчика. +
- b) Спецификации используются для проверки готовности ПО. +
- c) Спецификации являются неотъемлемой частью программной документации, облегчают сопровождение и модификацию ПО. +

3. Что происходит на второй стадии?

- a) Формируется структура ПО и разрабатываются алгоритмы, задаваемые спецификациями; +
- b) Устанавливается состав модулей с разделением их на иерархические уровни на основе изучения схем алгоритмов; +
- c) Выбирается структура информационных массивов; +
- d) Фиксируются межмодульные интерфейсы; +

4. В каких отношениях может находиться один класс с другим:

- a) Отношение наследования +
- b) Отношение включения +
- c) Отношение использования +

5. Отметьте только принципы использования цвета, которыми нужно руководствоваться при проектировании эргономичного интерфейса:

- a) для неактивных элементов лучше использовать бледные цвета; +
- b) выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем
- c) если надо разделить данные, то цвета выбирают из различных частей спектра: красный — зеленый, синий — желтый, любой цвет — белый +
- d) структура меню должна соответствовать структуре решаемой системой задачи; организация меню должна отразить наиболее эффективную последовательность шагов, ведущих к решению поставленной задачи
- e) при отображении состояния, как правило, красный означает опасность (стоп), зеленый — продолжение работы, желтый — цвет предупреждения +

6. Отметьте только принципы, которыми необходимо руководствоваться при создании текстовых диалогов:

- a) для неактивных элементов лучше использовать бледные цвета;

- b) выровненный по правому краю текст труднее читать, чем равномерно распределенный текст с невыровненным правым полем +
- c) текст в нижнем регистре читается приблизительно на 13% быстрее, чем текст, напечатанный полностью в верхнем регистре +
- d) пункты меню должны быть краткими, грамматически правильными и соответствовать своему заголовку. Порядок пунктов меню выбирается в соответствии с соглашением, частотой и порядком использования, а также в зависимости от потребностей задачи или пользователя
- e) для группировки данных, объединения и подбора нужно использовать соседние цвета спектра: оранжевые—желтые, синие—фиолетовые

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

1. Пользователь вводит набор целых чисел. Найти произведение нечетных элементов в указанном интервале и вывести его. Если таких элементов нет, то вывести сообщение об этом.

Требования к заданию:

- 1. Обязателен графический интерфейс.
- 2. 3 тестовых набора, описывающих разные ситуации.
- 3. Переменные и функции должны быть поименованы в соответствии с выполняемыми задачами.
- 4. Наличие комментариев к каждому блоку кода.
- 5. Презентация работы программы

```
from tkinter import *
from tkinter import ttk
# Импортируем модуль для создания интерфейса

# модуль расчета произведения нечетных чисел
def show_message():
    numbers = entry.get()
    list_number = [int(_) for _ in numbers.split()]
    array = []
    for number in list_number:
        if number % 2 != 0:
            array.append(number)
    result = 1
    for number in array:
        result *= number
    if result == 1:
        result = 'Нечетных элементов нет!'
    label["text"] = result # получаем введенный текст

# модуль создания и обработки интерфейса
root = Tk()
root.title('Произведение')
root.geometry("250x200")
entry = ttk.Entry()
entry.pack(anchor=NW, padx=9, pady=9)
btn = ttk.Button(text="Расчет", command=show_message)
btn.pack(anchor=NW, padx=9, pady=9)
label = ttk.Label()
label.pack(anchor=NW, padx=9, pady=9)
```

root.mainloop()

5. Зачётная ведомость.

2. Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которых также могут встречаться следующие символы:

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Определите, какое из указанных имён файлов удовлетворяет маске:

??pri*.*

- 1) napri.q
- 2) pripri.txt
- 3) privet.doc
- 4) 3priveta.c

Решение.

Символ «?» означает ровно один произвольный символ, значит, до «pri» должно быть не более двух символов. Этому условию удовлетворяет лишь один ответ: 1.

3. Сколько записей удовлетворяют условию «Пол = 'ж' ИЛИ Физика = 79»?

Фамилия	Пол	Математика	История	Физика	Химия	Биология
Андреев	м	80	72	68	66	70
Борисов	м	75	88	69	61	69
Васильева	ж	85	77	73	79	74
Дмитриев	м	77	85	81	81	80
Егорова	ж	88	75	79	85	75
Захарова	ж	72	80	66	70	70

Решение.

Логическое «ИЛИ» верно тогда, когда истинно хотя бы одно из утверждений.

Пол = 'ж' верно для Васильевой, Егоровой и Захаровой, Физика = '79' верно для Егоровой.

Логическая операция «ИЛИ» возвращает 0 только если оба аргумента имеют нулевое значение, следовательно, правильный ответ — 3.

Ответ: 3.

4. Во фрагменте базы данных представлены сведения о родственных отношениях. На основании приведённых данных определите ID дяди Керзона П. А. (дядя — это родной брат матери или отца).

Таблица 1		
ID	Фамилия_И.О	Пол
3272	Базилевич А. Б.	Ж
3228	Базилевич Б. Ф	М
3299	Базилевич И. Б.	М
3278	Базилевич П.И	М
3256	Базилевич Т. И.	Ж
3265	Керзон А. И.	Ж
3331	Керзон А. П.	М
3261	Керзон Л. А.	Ж
1217	Керзон П. А.	М
1202	Зельдович М. А.	Ж
3227	Витюк Д. А.	Ж
3240	Лемешко В. А.	Ж
3246	Месяц К. Г.	М
3387	Лукина Р. Г.	Ж
3293	Фокус П. А.	Ж
3332	Друк Г. Р.	Ж

Таблица 2	
ID_Родителя	ID_ребенка
3272	3272
3228	3299
3299	3272
3278	3299
3256	3240
3265	1202
3331	1217
3261	3356
1217	3378
1202	3356
3227	3378
3240	3240
3246	1202
3387	1217
3293	3261
3332	3293

Решение:

Из первой таблицы видно, что ID Керзона П. А. равен 1217. Найдем во второй таблице в графе «ID_ребенка» номер Керзона П. А. Видно, что его родители имеют ID 3272 и 3331. Родители обладателей этих ID имеют ID 3227 и 3228. Обладатели ID 3227 и 3228 имеют детей, ID которых 3299 и 3227.

Ответ: Значению ID 3299 соответствует Базилевич И. Б.

5. Дан рекурсивный алгоритм:

Найдите сумму чисел, которые будут выведены при вызове F(2).

```

procedure F(n: integer);
begin
  writeln(n);
  if n < 5 then begin
    F(n+1);
    F(n+2);
    F(n*3)
  end
end;

```

Ответ: 79

6. Дан рекурсивный алгоритм:

Чему будет равно значение, вычисленное алгоритмом при выполнении вызова F(5)?

```

function F(n: integer): integer;
begin
  if n > 2 then
    F := F(n - 1) + F(n - 2)
  else
    F := n;
  end;

```

Ответ: 8

7. Ниже записаны две рекурсивные функции F и G:

Чему будет равно значение, вычисленное при выполнении вызова F(6)?

```

function F(n: integer): integer;
begin
if n > 2 then
F := F(n - 1) + G(n - 2)
else
F := n;
end;
function G(n: integer): integer;
begin
if n > 2 then
G := G(n - 1) + F(n - 2)
else
G := n+1;
end;

```

Ответ: 17

8. Соотнесите понятия

A) Состояние объекта	1.некоторый объект или множество объектов
B) Поведение объекта	2.набор методов (программный код), оперирующих над состоянием объекта
C) Значение атрибута объекта	3.набор значений его атрибутов

A-3

B-2

C-1

9.Соотнести понятия спецификаторов:

Спецификаторы	Определения
A) private	1. защищенный, разрешено обращаться как с текущего класса, так и с классов наследников
B) protected	2. закрытый, то есть к нему можно обращаться только в текущем классе
C) public	3. общедоступный, разрешено обращаться из любого места программы

A-2

B-2

C-3

10. Соотнесите определения

A) Caption	1.Определяет общий вид окна и операции с ним, которые разрешается выполнять пользователю
Б) Icon	2. Определяет заголовок окна приложения
В) Icon	3. Определяет значок, который будет использоваться в строке заголовка формы при выполнении программы

A-2

B-3

C-1

11. Установите соответствие



А) размер (текста, символов)	1) эффективный метод для привлечения внимания к какой-либо части экрана
В) оттенение (различная текстура объектов)	2) обычно применяют увеличение выделенного объекта в 1,5 раза
С) окружение (подчеркивание, рамки, инвертированное изображение)	3) Очень эффективный способ, если не переусердствовать

А-2

В-2

С-3

12. Установите соответствие

а) форма (символ, шрифт, форма символа)	1) эффективный метод для привлечения внимания к какой-либо части экрана
б) размер (текста, символов)	2) используется для того, чтобы дифференцировать различные категории данных
с) оттенение (различная текстура объектов)	3) обычно применяют увеличение выделенного объекта в 1,5 раза

А-2

В-3

С-1

1.4 Контрольно-оценочные средства текущего контроля

МДК.01.02 Поддержка и тестирование программных модулей

Тема 1. Отладка и тестирование программного обеспечения

Вопросы с выбором одного варианта правильного ответа

1. Какие существуют особенности управляющего графа программы в случае объектно-ориентированного программирования (ООП)?

- а. она становится неприменимой
- б. она требует адаптации по обработке сообщений
- с. она требует описания поведения программы
- д. она требует описания не только структуры, но и поведения программы +

2. Какие существуют особенности интеграционного тестирования для ООП?

- а. тестирование дерева классов
- б. тестирование последовательностей прямых вызовов методов с помощью Р-путей
- с. тестирование последовательностей обработки сообщений с помощью ММ-путей
- д. тестирование исключительно цепочек вызовов процедур программ +

3. Какие этапы включает методика ООП при тестировании программного комплекса?

- а. тестирование методов каждого класса программного комплекса +
- б. тестирование отношений между классами с помощью тестов на основе Р-путей или ММ-путей
- с. тестирование взаимодействия модулей по всей иерархии комплекса

4. Какие этапы методики тестирования используются в ООП?

- a. тестирование методов каждого класса программы
- b. тестирование методов класса, входящих в его контекст +
- c. тестирование дерева классов программного проекта, включающего оттестированный класс

5. Какие этапы методов тестирования класса используются в ООП?

- a. тестирование класса как модуля по выбранному критерию
- b. тестирование класса как иерархической структуры +
- c. тестирование классов, входящих в модель проекта

6. Что такое управляющий граф программы (УГП)?

- a. множество операторов программы
- b. граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов +
- c. множество операторов управления

7. Что такое путь в УГП?

- a. последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной
- b. последовательность ветвей управляющего графа программы с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
- c. множество связанных дуг управляющего графа программы +

Вопрос 8. Что такое ветвь управляющего графа программы?

- a) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной, которые кодируют либо условные операторы, либо первый и последний операторы управляющего графа программы соответственно +
- b. часть пути, в котором все внутренние вершины кодируют линейные операторы
- c. начальная и конечная вершина пути

9. Можно ли гарантировать остановку программы на любом тесте?

- a. в общем случае нет +
- b. возможно в частных случаях
- c. задача в общей постановке алгоритмически неразрешима

10. Какие задачи у модульного тестирования?

- a. выявление локальных ошибок реализации алгоритмов модулей
- b. выявление ошибок при вызове модулей
- c. выявление ошибок взаимодействия модуля с окружением +

11. На основе каких принципов строятся тесты для модульного тестирования?

- a. анализ потоков управления модуля
- b. анализ потоков данных модуля +
- c. анализ покрытия в соответствии с заданными критериями C0, C1, C2

12. Каковы фазы процесса построения тестовых путей?

- a. построение управляющего графа программы
- b. выбор тестовых путей +
- c. генерация тестов, соответствующих выбранным тестовым путям

13. Какие существуют методы построения тестовых путей?

- a. статические

- b. динамические
- c. методы реализуемых путей +

14. Как реализуются динамические методы построения тестовых путей?

- a. наращивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие +
- b. построение пути методом удлинения за счет добавления дуг
- c. поиск всех реализуемых путей

15. Какие существуют разновидности интеграционного тестирования?

- a. монолитное тестирование +
- b. нисходящее тестирование
- c. восходящее тестирование
- d. Регрессионное тестирование

16. Каковы особенности нисходящего тестирования?

- a. необходимость разработки заглушек
- b. необходимость разработки среды управления очередностью вызовов модулей +
- c. параллельная разработка эффективных модулей

17. Каковы особенности восходящего тестирования?

- a. минимизация разработки заглушек
- b. запаздывание в проверке функциональности реализуемого приложения +
- c. необходимость разработки среды управления очередностью вызовов модулей

18. В чем заключаются особенности интеграционного тестирования для процедурного программирования?

- a. тестирование программных комплексов, заданных в виде иерархических структур модулей
- b. использование диаграмм потока управления в качестве модели тестируемого комплекса
- c. контроль соответствия спецификациям параметров модулей и межмодульных связей +
- d. контроль наследования

19. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена. К какой группе относится тест, покрывающий только функцию А?

- a. тесты, пригодные для повторного использования
- b. тесты, требующие повторного запуска +
- c. устаревшие тесты
- d. новые тесты

20. Возможно ли тестирование программы на всех допустимых значениях параметров?

- a. никогда +
- b. да, всегда
- c. возможно в отдельных случаях

Системное и регрессионное тестирование

1. Каковы особенности системного тестирования?

- a. тестированию подлежит система в целом
- b. тесты оперируют пользовательским или другими внешними интерфейсами +

- c. структура проекта тестируется на уровне подсистем
- d. тестирование осуществляется по методу «черного ящика»

2. Какие задачи решаются на уровне системного тестирования?

- a. выявление дефектов в функционировании приложения или в работе с ним
- b. выявление дефектов использования ресурсов
- c. выявление несовместимости с окружением
- d. выявление непредусмотренных сценариев применения +
- e. использования непредусмотренных комбинаций данных

3. Какие категории тестов разрабатываются для системного тестирования?

- a. тесты для проверки полноты функциональности
- b. тесты для проверки корректности использования ресурсов
- c. тесты для проверки стрессовых режимов и оценки производительности
- d. тесты для проверки защиты от искаженных данных и некорректных действий +
- e. тесты для проверки инсталляции и конфигурации для платформ, предусмотренных спецификацией
- f. тесты для проверки корректности пользовательской документации

4. Каковы особенности регрессионного тестирования?

- a. регрессионное тестирование является подмножеством системного тестирования
- b. выбор между полным и частичным перетестированием и пополнением тестовых наборов +
- c. перетестирование предусматривает только контроль частей приложения, связанных с изменениями

5. Какие задачи решает тестировщик, проверяя изменения, внесенные разработчиком в код?

- a. проверка и подтверждение исправления дефекта
- b. обеспечение гарантий, что качество исправленного кода не ухудшилось
- c. перетестирование последствий исправлений на предмет обнаружения индуцированных ошибок +
- d. проверка возможности воспроизведения ошибки каким-либо другим способом

6. Какие типы дефектов выявляются при системном или при регрессионном тестировании?

- a. отсутствующая или некорректная функциональность
- b. некорректность проектной документации
- c. непредусмотренные данные или неподдерживаемые сценарии использования +
- d. ошибки переносимости на другие платформы
- e. ошибки пользовательской документации
- f. ошибки инсталляции и конфигурирования

7. Можно ли гарантировать безопасность метода регрессионного тестирования при отсутствии информации об изменениях в программе

- a. нет +
- b. да

8. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?

- a. метод повторного прогона всех тестов
- b. случайные методы
- c. безопасные методы +
- d. методы минимизации
- e. методы, основанные на покрытии кода

9. Какими свойствами обладает метод повторного прогона всех тестов?

- a. полнота
- b. точность
- c. эффективность
- d. универсальность +

10. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий?

- a. тесты, пригодные для повторного использования
- b. тесты, требующие повторного запуска +
- c. устаревшие тесты
- d. новые тесты

11. Какие из перечисленных методов тестирования наиболее затратны

- a. системное тестирование с моделируемым окружением
- b. статические методы тестирования
- c. интеграционное тестирование +
- d. системное тестирование в реальном окружении и реальном времени
- e. модульное тестирование

12. Что влияет на обеспечение качества программного продукта?

- a. тестирование
- b. анализ дизайна
- c. обзоры кода
- d. аудиты процесса разработки ревьюирование проектной документации +
- e. ревьюирование проектной документации

13. Какие основные уровни выделены в процессе тестирования?

- a. модульное тестирование
- b. Регрессионное тестирование
- c. интеграционное тестирование +
- d. системное тестирование

14. Какие методы регрессионного тестирования применяются в условиях высоких требований к программному продукту?

- a. метод повторного прогона всех тестов
- b. случайные методы
- c. безопасные методы +
- d. методы, основанные на покрытии кода
- e. методы минимизации

15. В среднем, метод выборочного регрессионного тестирования...

- a. обнаруживает столько же ошибок, сколько и метод повторного прогона всех тестов
- b. обнаруживает больше ошибок, чем метод повторного прогона всех тестов +
- c. обнаруживает меньше ошибок, чем метод повторного прогона всех тестов
- d. может обнаруживать больше или меньше ошибок, чем метод повторного прогона всех тестов

16. На каком этапе регрессионного тестирования удаляются устаревшие тесты?

- a. предсказание целесообразности
- b. идентификация изменений
- c. отбор тестов +
- d. выполнение тестов
- e. создание дополнительных тестов
- f. обновление базы данных

17. На каком этапе регрессионного тестирования проводится упорядочение тестов?

- a. предсказание целесообразности
- b. идентификация изменений
- c. отбор тестов
- d. выполнение тестов +
- e. создание дополнительных тестов
- f. обновление базы данных

Задания с выбором нескольких вариантов правильных ответов

1. Как реализуются динамические методы построения тестовых путей?

- a) Поиск всех реализуемых путей +
- b) Наращивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие +
- c) Построение пути методом удлинения за счет добавления дуг

2. Какие существуют разновидности интеграционного тестирования?

- a) Регрессионное тестирование
- b) Восходящее тестирование +
- c) Нисходящее тестирование +
- d) Монолитное тестирование +

3. Какая информация должна сопровождать действие по исправлению ошибки и перевод дефекта в состояние Resolved?

- a) Краткий комментарий сделанных исправлений +
- b) Причину возникновения дефекта +
- c) Место исправления дефекта +

Тема 2. Документирование

Вопросы с выбором одного варианта правильного ответа

1. Является ли программа аналогом математической формулы?

- a) Да +
- b) Нет
- c) Математические формулы и программы не сводятся друг к другу

2. Какова мощность множества тестов, формально необходимая для тестирования операции в машине с 32-разрядным машинным словом?

- a) 2^{32} +
- b) 4^9
- c) 2^{64}

3. Сколько тестов потребуется для проверки программы, реализующей задержку на неопределенное количество тактов?

- a) Один
- b) Неопределенное количество
- c) Зависит от критерия достаточности проверок +

4. Зачем нужна спецификация тестирования?

- a) Для формирования команды тестировщиков
- b) Для разработки тестового набора +
- c) Для понимания смысла программы

5. Зачем нужен Log-файл?

- a) Для изучения результатов тестирования в режиме on-line
- b) Для фиксации результатов прогона test-suite +

Вопросы с выбором нескольких вариантов правильных ответов

6. Какие подходы используются для обоснования истинности программ?

- a) Использование аналогий
- b) Эксперимент над программой +
- c) Доказательство программы +
- d) Формальный и интерпретационный +

7. Отметьте верные утверждения

- a) Тестирование – процесс поиска ошибок +
- b) В фазу тестирования входят поиски и исправление ошибок +
- c) Отладка – процесс локализации и исправления ошибок +

8. Какие существуют методы анализа и локализации ошибки?

- a) Выполнение программы в уме +
- b) Пошаговое выполнение +
- c) Метод контрольных точек и анализа трасс +

1.5 Контрольно-оценочные средства для дифференцированного зачета

Вопросы с выбором одного варианта правильного ответа

1. Какие существуют особенности управляющего графа программы в случае объектно-ориентированного программирования (ООП)?

- a. она становится неприменимой
- b. она требует адаптации по обработке сообщений
- c. она требует описания поведения программы
- d. она требует описания не только структуры, но и поведения программы +

2. Какие существуют особенности интеграционного тестирования для ООП?

- a. тестирование дерева классов
- b. тестирование последовательностей прямых вызовов методов с помощью Р-путей
- c. тестирование последовательностей обработки сообщений с помощью ММ-путей
- d. тестирование исключительно цепочек вызовов процедур программ +

3. Какие этапы включает методика ООП при тестировании программного комплекса?

- a. тестирование методов каждого класса программного комплекса +
- b. тестирование отношений между классами с помощью тестов на основе Р-путей или ММ-путей
- c. тестирование взаимодействия модулей по всей иерархии комплекса

4. Какие этапы методики тестирования используются в ООП?

- a. тестирование методов каждого класса программы
- b. тестирование методов класса, входящих в его контекст +
- c. тестирование дерева классов программного проекта, включающего оттестированный класс

5. Какие этапы методов тестирования класса используются в ООП?

- a. тестирование класса как модуля по выбранному критерию
- b. тестирование класса как иерархической структуры +

с. тестирование классов, входящих в модель проекта

6. Что такое управляющий граф программы (УГП)?

- а. множество операторов программы
- б. граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов +
- с. множество операторов управления

7. Что такое путь в УГП?

- а. последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной
- б. последовательность ветвей управляющего графа программы с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
- с. множество связанных дуг управляющего графа программы +

Вопрос 8. Что такое ветвь управляющего графа программы?

- а) последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной, которые кодируют либо условные операторы, либо первый и последний операторы управляющего графа программы соответственно +
- б. часть пути, в котором все внутренние вершины кодируют линейные операторы
- с. начальная и конечная вершина пути

9. Можно ли гарантировать остановку программы на любом тесте?

- а. в общем случае нет +
- б. возможно в частных случаях
- с. задача в общей постановке алгоритмически неразрешима

10. Какие задачи у модульного тестирования?

- а. выявление локальных ошибок реализации алгоритмов модулей
- б. выявление ошибок при вызове модулей
- с. выявление ошибок взаимодействия модуля с окружением +

11. На основе каких принципов строятся тесты для модульного тестирования?

- а. анализ потоков управления модуля
- б. анализ потоков данных модуля +
- с. анализ покрытия в соответствии с заданными критериями C0, C1, C2

12. Каковы фазы процесса построения тестовых путей?

- а. построение управляющего графа программы
- б. выбор тестовых путей +
- с. генерация тестов, соответствующих выбранным тестовым путям

13. Какие существуют методы построения тестовых путей?

- а. статические
- б. динамические
- с. методы реализуемых путей +

14. Как реализуются динамические методы построения тестовых путей?

- а. наращивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие +
- б. построение пути методом удлинения за счет добавления дуг
- с. поиск всех реализуемых путей

15. Какие существуют разновидности интеграционного тестирования?

- а. монолитное тестирование +
- б. нисходящее тестирование
- с. восходящее тестирование

d. Регрессионное тестирование

16. Каковы особенности нисходящего тестирования?

- a. необходимость разработки заглушек
- b. необходимость разработки среды управления очередностью вызовов модулей +
- c. параллельная разработка эффективных модулей

17. Каковы особенности восходящего тестирования?

- a. минимизация разработки заглушек
- b. запаздывание в проверке функциональности реализуемого приложения +
- c. необходимость разработки среды управления очередностью вызовов модулей

18. В чем заключаются особенности интеграционного тестирования для процедурного программирования?

- a. тестирование программных комплексов, заданных в виде иерархических структур модулей
- b. использование диаграмм потока управления в качестве модели тестируемого комплекса
- c. контроль соответствия спецификациям параметров модулей и межмодульных связей +
- d. контроль наследования

19. При создании очередной версии программы была добавлена функция А, функция D была удалена, функция С – изменена. К какой группе относится тест, покрывающий только функцию А?

- a. тесты, пригодные для повторного использования
- b. тесты, требующие повторного запуска +
- c. устаревшие тесты
- d. новые тесты

20. Возможно ли тестирование программы на всех допустимых значениях параметров?

- a. никогда +
- b. да, всегда
- c. возможно в отдельных случаях

Системное и регрессионное тестирование

1. Каковы особенности системного тестирования?

- a. тестированию подлежит система в целом
- b. тесты оперируют пользовательским или другими внешними интерфейсами +
- c. структура проекта тестируется на уровне подсистем
- d. тестирование осуществляется по методу «черного ящика»

2. Какие задачи решаются на уровне системного тестирования?

- a. выявление дефектов в функционировании приложения или в работе с ним
- b. выявление дефектов использования ресурсов
- c. выявление несовместимости с окружением
- d. выявление непредусмотренных сценариев применения +
- e. использования непредусмотренных комбинаций данных

3. Какие категории тестов разрабатываются для системного тестирования?

- a. тесты для проверки полноты функциональности
- b. тесты для проверки корректности использования ресурсов
- c. тесты для проверки стрессовых режимов и оценки производительности
- d. тесты для проверки защиты от искаженных данных и некорректных действий +
- e. тесты для проверки инсталляции и конфигурации для платформ, предусмотренных спецификацией
- f. тесты для проверки корректности пользовательской документации

4. Каковы особенности регрессионного тестирования?

- a. регрессионное тестирование является подмножеством системного тестирования
- b. выбор между полным и частичным перетестированием и пополнением тестовых наборов +
- c. перетестирование предусматривает только контроль частей приложения, связанных с изменениями

5. Какие задачи решает тестировщик, проверяя изменения, внесенные разработчиком в код?

- a. проверка и подтверждение исправления дефекта
- b. обеспечение гарантий, что качество исправленного кода не ухудшилось
- c. перетестирование последствий исправлений на предмет обнаружения индуцированных ошибок +
- d. проверка возможности воспроизведения ошибки каким-либо другим способом

6. Какие типы дефектов выявляются при системном или при регрессионном тестировании?

- a. отсутствующая или некорректная функциональность
- b. некорректность проектной документации
- c. непредусмотренные данные или неподдерживаемые сценарии использования +
- d. ошибки переносимости на другие платформы
- e. ошибки пользовательской документации
- f. ошибки инсталляции и конфигурирования

7. Можно ли гарантировать безопасность метода регрессионного тестирования при отсутствии информации об изменениях в программе

- a. нет +
- b. да

8. Какие методы регрессионного тестирования применяются в условиях высоких требований к качеству программного продукта?

- a. метод повторного прогона всех тестов
- b. случайные методы
- c. безопасные методы +
- d. методы минимизации
- e. методы, основанные на покрытии кода

9. Какими свойствами обладает метод повторного прогона всех тестов?

- a. полнота
- b. точность
- c. эффективность
- d. универсальность +

10. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия тестов, использовавшихся при тестировании предыдущих версий?

- a. тесты, пригодные для повторного использования
- b. тесты, требующие повторного запуска +
- c. устаревшие тесты
- d. новые тесты

11. Какие из перечисленных методов тестирования наиболее затратны

- a. системное тестирование с моделируемым окружением
- b. статические методы тестирования
- c. интеграционное тестирование +
- d. системное тестирование в реальном окружении и реальном времени
- e. модульное тестирование

12. Какие из перечисленных методов тестирования дают наиболее надежные результаты?

- a. статические методы
- b. модульное тестирование
- c. интеграционное тестирование
- d. системное тестирование с моделируемым окружением +

е. системное тестирование в реальном окружении и реальном времени

13. Что влияет на обеспечение качества программного продукта?

- а. тестирование
- б. анализ дизайна
- в. обзоры кода
- г. аудиты процесса разработки ревьюирование проектной документации +
- е. ревьюирование проектной документации

14. Какие основные уровни выделены в процессе тестирования?

- а. модульное тестирование
- б. Регрессионное тестирование
- в. интеграционное тестирование +
- г. системное тестирование

15. Какие методы регрессионного тестирования применяются в условиях высоких требований к программному продукту?

- а. метод повторного прогона всех тестов
- б. случайные методы
- в. безопасные методы +
- г. методы, основанные на покрытии кода
- е. методы минимизации

16. В среднем, метод выборочного регрессионного тестирования...

- а. обнаруживает столько же ошибок, сколько и метод повторного прогона всех тестов
- б. обнаруживает больше ошибок, чем метод повторного прогона всех тестов +
- в. обнаруживает меньше ошибок, чем метод повторного прогона всех тестов
- г. может обнаруживать больше или меньше ошибок, чем метод повторного прогона всех тестов

17. На каком этапе регрессионного тестирования удаляются устаревшие тесты?

- а. предсказание целесообразности
- б. идентификация изменений
- в. отбор тестов +
- г. выполнение тестов
- е. создание дополнительных тестов
- ф. обновление базы данных

18. На каком этапе регрессионного тестирования проводится упорядочение тестов?

- а. предсказание целесообразности
- б. идентификация изменений
- в. отбор тестов
- г. выполнение тестов +
- е. создание дополнительных тестов
- ф. обновление базы данных

19. Является ли программа аналогом математической формулы?

- а) Да +
- б) Нет
- в) Математические формулы и программы не сводятся друг к другу

20. Какова мощность множества тестов, формально необходимая для тестирования операции в машине с 32-разрядным машинным словом?

- а) 2^{32} +
- б) 4^9
- в) 2^{64}

21. Сколько тестов потребуется для проверки программы, реализующей задержку на

неопределенное количество тактов?

- a) Один
- b) Неопределенное количество
- c) Зависит от критерия достаточности проверок +

22.Зачем нужна спецификация тестирования?

- a) Для формирования команды тестирующих
- b) Для разработки тестового набора +
- c) Для понимания смысла программы

23.Зачем нужен Log-файл?

- a) Для изучения результатов тестирования в режиме on-line
- b) Для фиксации результатов прогона test-suite +
- c) Для записи комментариев после прогона тестов

Вопросы с выбором нескольких вариантов правильных ответов

1.Как реализуются динамические методы построения тестовых путей?

- a) Поиск всех реализуемых путей +
- b) Нарастивание начальных отрезков реализованных путей продолжающими их фрагментами, чтобы увеличить покрытие +
- c) Построение пути методом удлинения за счет добавления дуг

2.Какие существуют разновидности интеграционного тестирования?

- a) Регрессионное тестирование
- b) Восходящее тестирование +
- c) Нисходящее тестирование +
- d) Монолитное тестирование +

3.Какая информация должна сопровождать действие по исправлению ошибки и перевод дефекта в состояние Resolved?

- a) Краткий комментарий сделанных исправлений +
- b) Причину возникновения дефекта +
- c) Место исправления дефекта +

4.Какие подходы используются для обоснования истинности программ?

- a) Использование аналогий
- b) Эксперимент над программой +
- c) Доказательство программы +
- d) Формальный и интерпретационный +

5.Отметьте верные утверждения

- a) Тестирование – процесс поиска ошибок +
- b) В фазу тестирования входят поиски и исправление ошибок +
- c) Отладка – процесс локализации и исправления ошибок +

6.Какие существуют методы анализа и локализации ошибки?

- a) Выполнение программы в уме +
- b) Пошаговое выполнение +
- c) Метод контрольных точек и анализа трасс +

1.6 Контрольно- оценочные средства для проведения текущего контроля знаний

Перечень тем (разделов):

1. Основные платформы мобильных приложений, сравнительная характеристика
2. Нативные приложения, веб-приложения, гибридные и кроссплатформенные приложения, их области применения
3. Основные языки для разработки мобильных приложений (Java, ObjectiveC и др.)
4. Инструменты разработки мобильных приложений (JDK/AndroidStudio/WebView/Phonegap и др.)
5. Инструментарий среды разработки мобильных приложений. Структура типичного мобильного приложения.
6. Архитектура платформы Android. Dalvik Virtual Machine. Среда разработки для Android. Eclipse IDE. П
7. Плагин ADT. Android Virtual Device.
8. Android SDK. Версии SDK и Android API Level.
9. Инструментарий среды разработки мобильных приложений
10. Структура типичного мобильного приложения
11. Элементы управления и контейнеры
12. Работа со списками
13. Способы хранения данных

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
<i>Уметь:</i>	
Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.
Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9, 10; ПК 1.1, 1.2.	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма.
Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9, 10; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля
Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9, 10; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования
Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9, 10; ПК 1.4, 1.5	Производит оптимизацию программного кода и его рефакторинг.
Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9, 10; 11 ПК 1.2	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства

Вопросы с выбором одного варианта правильного ответа

1. Что такое мобильное приложение?

- а) Программа, которая устанавливается на мобильное устройство и выполняет определенные функции. +
- б) Веб-приложение, которое работает только в браузере.
- в) Приложение, которое работает только на настольных компьютерах.

2. Какие языки программирования используются для разработки мобильных приложений?

- а) JavaScript, Python, Java, Swift. +
- б) JavaScript, Python, C++, Swift.
- в) JavaScript, Python, Java, C++.

3. Что такое React Native?

- a) Фреймворк для разработки мобильных приложений на JavaScript. +
- b) Библиотека для создания веб-приложений.
- c) Язык программирования для создания игр.

4. Какие преимущества имеет разработка мобильных приложений на JavaScript?

- b) Высокая скорость разработки, возможность быстрого прототипирования.
- c) Легкость интеграции с веб-сервисами и базами данных.
- a) Кроссплатформенность, возможность использования одного кода для разных платформ. +

5. Какие инструменты используются для разработки мобильных приложений на JavaScript?

- a) React Native, Ionic, Cordova. +
- b) React Native, Angular, Vue.js.
- c) React Native, Node.js, Express.js.

6. Какие функции можно реализовать в мобильном приложении на JavaScript?

- a) Работа с камерой, GPS, уведомлениями. +
- b) Работа с базами данных, веб-сервисами.
- c) Работа с графикой, анимацией.

7. Какие преимущества имеет React Native перед нативными мобильными приложениями?

- a) Более быстрая разработка, возможность использования одного кода для разных платформ. +
- b) Более низкая стоимость разработки, возможность использования одного кода для разных платформ.
- c) Более легкая интеграция с веб-сервисами и базами данных.

8. Какие платформы поддерживает React Native?

- a) iOS и Android. +
- b) iOS, Android, Windows Phone.
- c) iOS, Android, BlackBerry.

9. Какие преимущества имеет разработка мобильных приложений на JavaScript перед разработкой нативных приложений?

- a) Более быстрая разработка, возможность использования одного кода для разных платформ. +
- b) Более низкая стоимость разработки, возможность использования одного кода для разных платформ.
- c) Более легкая интеграция с веб-сервисами и базами данных.

10. Какие инструменты используются для тестирования мобильных приложений на JavaScript?

- a) Jest, Enzyme, Detox. +
- b) Mocha, Chai, Sinon.
- c) Karma, Jasmine, Protractor.

11. Какие инструменты используются для сборки и развертывания мобильных приложений на JavaScript?

- a) npm, yarn, webpack. +
- b) npm, yarn, gulp.
- c) npm, yarn, grunt.

12. Какие преимущества имеет разработка мобильных приложений на JavaScript перед разработкой нативных приложений?

- a) Более быстрая разработка, возможность использования одного кода для разных платформ. +
- b) Более низкая стоимость разработки, возможность использования одного кода для разных платформ.

с) Более легкая интеграция с веб-сервисами и базами данных.

13. Какие инструменты используются для создания пользовательского интерфейса в мобильных приложениях на JavaScript?

- a) React Native, Ionic, Cordova. +
- b) React Native, Angular, Vue.js.
- c) React Native, Node.js, Express.js.

14. Что такое JavaScript?

- a) Язык программирования, используемый для создания веб-страниц. +
- b) Язык программирования, используемый для создания мобильных приложений.
- c) Язык программирования, используемый для создания игр.

15. Какие основные принципы лежат в основе JavaScript?

- a) Объектно-ориентированное программирование. +
- b) Функциональное программирование.
- c) Прототипное программирование.

16. Какие основные конструкции языка JavaScript?

- a) Переменные, функции, операторы, циклы. +
- b) Классы, методы, свойства, наследование.
- c) Объекты, массивы, прототипы, замыкания.

17. Какие возможности предоставляет JavaScript для работы с веб-страницами?

- a) Изменение стилей элементов на странице. +
- b) Обработка событий на странице.
- c) Создание и манипуляция элементами на странице.

18. Какие библиотеки и фреймворки могут использоваться вместе с JavaScript для разработки веб-приложений?

- a) React, Angular, Vue.js. +
- b) jQuery, MooTools, Prototype.
- c) Node.js, Express.js, Socket.io.

Вопросы с выбором нескольких вариантов правильных ответов

1. Какие методы отражают жизненный цикл Activity:

- a. onCreate() (+)
- b. onRestart() (+)
- c. onResume() (+)
- d. onRunning()
- e. onClose()

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

Задание 1. Скачать Android SDK + Eclipse (Eclipse ADT Bundle) <http://developer.android.com/sdk/index.html>, а также последнюю версию Java (JDK) <http://www.oracle.com/technetwork/java/index-jsp-138363.html> (или просто набрать в yandex "JDK" и пройти по ссылке на oracle.com). Установить всё это. Создать новый проект, зайти в Android SDK и скачать какую-нибудь версию Android SDK (например, lollipop - 5.0.1),

Задание 2. Создать приложение, содержащее анимированные интерфейсные элементы (например, увеличивающиеся при клике на них кнопки, вращающиеся TextView и т.д.).

Задание 3. Создать приложение, отображающее после запуска карты Google или какие-нибудь другие карты.

Задание 4. Создать приложение, использующее опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента. Естественно, выбор пунктов меню должен что-то менять в интерфейсных элементах или их отображении! Например, очистить поле ввода через контекстное меню, или отобразить невидимые интерфейсные элементы через установку галок в опциональном меню.

Задание 5. Создать приложение, отображающее после некоторых действий (нажатия на кнопку, например, или проверки корректности ввода текста в EditText) диалоговое окно, свидетельствующее об ошибке/информирующее/предупреждающее пользователя о чём-то.

Задание 6. Создать приложение, помещающее по нажатию на кнопку какое-то сообщение со звуком в панель уведомлений/статус-панель на эмуляторе.

Задание 7. Создать приложение, по нажатию кнопки в котором проигрывается какой-то звук.

Задание 7. Создать приложение, при запуске которого проигрывается какое-то видео.

Дать развернутый ответ на вопрос

1. Каковы основные этапы разработки мобильного приложения?

Ответ:

создание концепции и проектирование, разработка и тестирование, публикация и поддержка

2. Какие основные компоненты включает в себя мобильное приложение?

Ответ:

пользовательский интерфейс, базу данных, серверные компоненты

3. Какие технологии обычно используются при разработке мобильных приложений?

Ответ:

iOS, Android, React Native, Flutter, Cordova, Ionic, Unity.

4. Что такое кроссплатформенная разработка мобильных приложений и каковы ее преимущества?

Ответ:

возможность разработки на различных платформах с использованием одной кодовой базы; снижение затрат на разработку; удобство использования для пользователей,

5. Какие виды монетизации мобильных приложений существуют?

Ответ:

реклама, подписка, внутренняя монетизация, интеграция со сторонними сервисами.

1.7 Контрольно-оценочные средства для проведения промежуточной аттестации

Вопросы с выбором одного варианта правильного ответа

1. Набор средств программирования, который содержит инструменты, необходимые для создания, компиляции и сборки мобильного приложения называется:

- а) Android SDK +
- б) JDK
- в) плагин ADT
- г) Android NDK

2. С какой целью был создан Open Handset Alliance?

- а) писать историю развития ОС Android
- б) продавать смартфоны под управлением Android
- в) рекламировать смартфоны под управлением Android
- г) разрабатывать открытые стандарты для мобильных устройств +

3. С какой целью инструмент Intel* Graphics Performance Analyzers (Intel* GPA) System Analyzer используется в среде разработки Intel* Beacon Mountain?

- а) позволить разработчикам оптимизировать загрузженность системы при использовании процедур OpenGL +
- б) для ускорения работы эмулятора в среде разработки
- в) для оптимизированной обработки данных и изображений
- г) позволить разработчикам эффективно распараллелить C++ мобильные приложения

4. Библиотеки, реализованные на базе PacketVideo OpenCORE:

- а) Media Framework +
- б) SQLite
- в) FreeType
- г) 3D библиотеки

5. Какой движок баз данных используется в ОС Android?

- а) InnoDB
- б) DBM
- в) MyISAM
- г) SQLite +

6. С какой целью инструмент Intel* Integrated Performance Primitives (Intel* IPP) используется в среде разработки Intel* Beacon Mountain?

- а) для оптимизированной обработки данных и изображений +
- б) позволить разработчикам оптимизировать загрузженность системы при использовании процедур OpenGL
- в) для ускорения работы эмулятора в среде разработки
- г) позволить разработчикам эффективно распараллелить C++ мобильные приложения

7. Intel XDK поддерживает разработку под:

- а) JavaFX Mobile
- б) Apple iOS, BlackBerry OS
- в) MtkOS, Symbian OS, Microsoft Windows 8
- г) Android, Apple iOS, Microsoft Windows 8, Tizen +

8. Каждый приемник широковещательных сообщений является наследником класса...

- а) ViewReceiver
- б) IntentReceiver
- в) ContentProvider
- г) BroadcastReceiver +

9. Какой класс является основным строительным блоком для компонентов пользовательского интерфейса (UI), определяет прямоугольную область экрана и отвечает за прорисовку и обработку событий?

- а) GUI
- б) View +

- в) UIComponent
- г) Widget

10. Какой слушатель используется для отслеживания события касания экрана устройства?

- а) OnPressListener
- б) onTouchListener +
- в) OnClickListener
- г) OnInputListener

11. В какой папке необходимо разместить XML файлы, которые определяют все меню приложения?

- а) res/value
- б) res/items
- в) res/layout
- г) res/menu +

12. Фоновые приложения ...

- а) после настройки не предполагают взаимодействия с пользователем, большую часть времени находятся и работают в скрытом состоянии +
- б) выполняют свои функции и когда видимы на экране, и когда скрыты другими приложениями
- в) небольшие приложения, отображаемые в виде графического объекта на рабочем столе
- г) большую часть времени работают в фоновом режиме, однако допускают взаимодействие с пользователем и после настройки

13. Какой компонент управляет распределенным множеством данных приложения?

- а) сервис (Service)
- б) активность (Activity)
- в) приемник широковестьных сообщений (Broadcast Receiver)
- г) контент-провайдер (Content Provider) +

14. Какой язык разметки используется для описания иерархии компонентов графическогопользовательского интерфейса Android-приложения?

- а) html
- б) xml +
- в) gml
- г) xhtml

15. Выберите верную последовательность действий, необходимых для создания в приложенииконтент-провайдера.

- а) Создание класса наследника от класса ContentProvider; Определение способа организации данных; Заполнение контент-провайдера данными
- б) Проектирование способа хранения данных; Определение способа организации данных;
- в) Создание класса наследника от класса ContentProvider; Заполнение контент-провайдера данными; Определение способа работы с данными
- г) Проектирование способа хранения данных; Создание класса наследника от класса ContentProvider; Определение строки авторизации провайдера, URI для его строк и имен столбцов +

16. Какие элементы управления применяются для действий по настройке?

- а) командные элементы управления
- б) элементы выбора +
- в) элементы ввода
- г) элементы отображения

17. Примерами комбо-элементов не являются:

- а) комбо-список
- б) все вышеперечисленное
- в) комбо-кнопка
- г) комбо-поле +

18. Более крупные элементы:

- а) привлекают больше внимания +
- б) все варианты ответа верны
- в) размер не влияет на уровень внимания
- г) привлекают меньше внимания

19. К традиционным типографическим инструментам не относят

- а) масштаб
- б) цвет +
- в) разреженность
- г) выравнивание по сетке

20. К элементам ввода относят:

- а) ограничивающие элементы ввода +
- б) ползунки +
- в) счетчики +

21. Выделяют следующие категории плотности экрана для Android-устройств:

- а) HDPI, XHDPI, XXHDPI, и XXXHDPI
- б) правильный вариант ответа отсутствует
- в) LDPI, MDPI, HDPI, XHDPI, XXHDPI, и XXXHDPI +
- г) LDPI, MDPI, HDPI

22. Основные вкладки (FixedTabs) удобны при отображении

- а) от четырех вкладок
- б) двух вкладок
- в) трех и более вкладок
- г) трех и менее вкладок +

23. Диалоговое окно, содержащее линейку процесса выполнения какого-то действия – это

- а) DatePickerDialog
- б) AlertDialog
- в) ProgressDialog +
- г) DialogFragment

24. Уведомления стоит использовать, когда

- а) сообщение не требует ответа пользователя, но важно для продолжения его работы
- б) сообщение является важным и требует немедленного прочтения и ответа
- в) сообщение является важным, однако требует немедленного прочтения, но не ответа
- г) сообщение является важным, однако не требует немедленного прочтения и ответа +

25. Какой метод запускает новую активность?

- А) startActivity () +
- б) beginActivity ()
- в) intentActivity ()
- г) newActivity ()

Вопросы с выбором нескольких вариантов правильных ответов

1. Полный иерархический список обязательных файлов и папок проекта можно увидеть на вкладке ...

- а) Package Explorer +
- б) Internet Explorer
- в) Navigator

г) Project Explorer +

2. Выберите верные утверждения относительно объекта-намерения (Intent).

- а) представляет собой структуру данных, содержащую описание операции, которая должна быть выполнена, и обычно используется для запуска активности или сервиса +
- б) используется для передачи сообщений пользователю
- в) используется для получения инструкций от пользователя
- г) используются для передачи сообщений между основными компонентами приложений +

3. Расположение элементов мобильного приложения:

- а) полезно для передачи иерархии +
- б) влияет на удобство использования +
- в) полезно для создания пространственных отношений между объектами на экране и объектами реального мира +

4. Дизайн или проектирование интерфейса для графических дизайнеров:

- а) прозрачность и понятность информации +
- б) тон, стиль, композиция, которые являются атрибутами бренда +
- в) передача информации о поведении посредством ожидаемого назначения +

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

Задание 1. Скачать Android SDK + Eclipse (Eclipse ADT Bundle)

<http://developer.android.com/sdk/index.html>, а также последнюю версию Java (JDK) <http://www.oracle.com/technetwork/java/index-jsp-138363.html> (или просто набрать в yandex "JDK" и пройти по ссылке на oracle.com). Установить всё это. Создать новый проект, зайти в Android SDK и скачать какую-нибудь версию Android SDK (например, lollipop - 5.0.1),

Задание 2. Создать приложение, содержащее анимированные интерфейсные элементы (например, увеличивающиеся при клике на них кнопки, вращающиеся TextView и т.д.).

Задание 3. Создать приложение, отображающее после запуска карты Google или какие-нибудь другие карты.

Задание 4. Создать приложение, использующее опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента. Естественно, выбор пунктов меню должен что-то менять в интерфейсных элементах или их отображении! Например, очистить поле ввода через контекстное меню, или отобразить невидимые интерфейсные элементы через установку галок в опциональном меню.

Задание 5. Создать приложение, отображающее после некоторых действий (нажатия на кнопку, например, или проверки корректности ввода текста в EditText) диалоговое окно, свидетельствующее об ошибке/информирующее/предупреждающее пользователя о чём-то.

Задание 6. Создать приложение, помещающее по нажатию на кнопку какое-то сообщение со звуком в панель уведомлений/статус-панель па эмуляторе.

Задание 7. Создать приложение, по нажатию кнопки в котором проигрывается какой-то звук.

Задание 7. Создать приложение, при запуске которого проигрывается какое-то видео.

Дать развернутый ответ на вопрос

1. Каковы основные этапы разработки мобильного приложения?

Ответ:

создание концепции и проектирование, разработка и тестирование, публикация и поддержка

2. Какие основные компоненты включает в себя мобильное приложение?

Ответ:

пользовательский интерфейс, базу данных, серверные компоненты

3. Какие технологии обычно используются при разработке мобильных приложений?

Ответ:

iOS, Android, React Native, Flutter, Cordova, Ionic, Unity₂

4. Что такое кроссплатформенная разработка мобильных приложений и каковы ее преимущества?

Ответ:

возможность разработки на различных платформах с использованием одной кодовой базы; снижение затрат на разработку; удобство использования для пользователей,

5. Какие виды монетизации мобильных приложений существуют?

Ответ:

реклама, подписка, внутренняя монетизация, интеграция со сторонними сервисами.

МДК 01.04 Системное программирование

1.8 Контрольно-оценочные средства для проведения промежуточной аттестации

Перечень тем, выносимых на тестирование:

1. Программирование на языке низкого уровня:

подсистемы управления ресурсами;
управление процессами;
управление потоками;
параллельная обработка потоков;
создание процессов и потоков;
обмен данными между процессами. Передача сообщений;
анонимные и именованные каналы;
сетевое программирование сокетов;
динамически подключаемые библиотеки DLL;
сервисы;
виртуальная память. Выделение памяти процессам;
работа с буфером экрана

Оценка запланированных результатов по МДК

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата
Уметь	
Осуществлять разработку кода программного модуля на языках низкого и высокого уровней; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля.

Создавать программу по разработанному алгоритму как отдельный модуль; ОК 1, 2, 3, 5, 9; ПК 1.1, 1.2	Разрабатывает оптимальный алгоритм решения задачи; создает программу на основе этого алгоритма
Выполнять отладку и тестирование программы на уровне модуля; ОК 1, 2, 3, 5, 9; ПК 1.3, 1.4.	Определяет и создает оптимальные тестовые кейсы для программного модуля; выполняет отладку и тестирование программы на уровне модуля
Осуществлять разработку кода программного модуля на современных языках программирования; ОК 1, 2, 3, 5, 9; ПК 1.2.	Определяет необходимый язык программирования; проводит разработку кода программного модуля на современном языке программирования.
Уметь выполнять оптимизацию и рефакторинг программного кода; ОК 1, 2, 3, 5, 9; ПК 1.4, 1.5.	Производит оптимизацию программного кода и его рефакторинг.
Оформлять документацию на программные средства. ОК 1, 2, 3, 4, 5, 6, 7, 9; 11 ПК 1.2.	Применяет стандарты оформления документации; грамотно оформляет документацию на программные средства.

1 Установите порядок, в котором осуществляется перевод произвольного двоичного числа в системе счисления с основанием $q=2n$. Укажите порядок следования всех 3 вариантов ответа:

1 рассмотреть каждую группу как n -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием $q = 2n$;

2 двоичное число разбить слева и справа (целую и дробную части) на группы по цифр в каждой;

3 если в последних правой и левой группах окажется меньше разрядов, то их надо дополнить справа и слева нулями до нужного числа разрядов

- a. 3 1 2
- b. 2 1 3
- c. 1 2 3 +
- d. 3 2 1

2 По сколько цифр в группе нужно разбивать двоичное число при переводе его в четверичную систему счисления?

- a. 6
- b. 2 +
- c. 3
- d. 4

3 Переведите двоичное число 10001111011 в шестандцатеричную систему счисления:

- a. 5fa +
- b. 8fb
- c. 8fc
- d. 8fa

4 Чему будет равен результат сложение двоичных чисел 1001 и 1000

- a. 10001
- b. 11001
- c. 10011
- d. 10101 +

5 Выполните деление в двоичной системе счисления $1110:10=$

- a. 101
- b. 011
- c. 110
- d. 111 +

6 Переведите двоичное число 1010101 в восьмеричную систему счисления

- a. 125
- b. 124 +
- c. 126
- d. 123

7 Установите порядок, в котором осуществляется перевод целого числа в системе счисления с основанием $q=2n$. Укажите порядок следования всех 3 вариантов ответа:

1 данное двоичное число разбить справа налево на группы по n цифр в каждой;

2 рассмотреть каждую группу как n -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием $q = 2n$;

3 если в последней левой группе окажется меньше разрядов, то ее надо дополнить слева нулями до нужного числа разрядов

- a. 1 3 2

- b. 1 2 3
- c. 3 2 1
- d. 2 1 3 +

8 Переведите шестнадцатеричное число FACC в двоичную систему счисления

- a. 1111001011011001
- b. 1111101011001100
- c. 1011010111111100
- d. 1101100111011010 +

9 Выполните вычисление в двоичной системе счисления $110101-101=$

- a. 110100 +
- b. 110010
- c. 110000
- d. 110001

10 В саду 100 фруктовых деревьев - 14 яблонь и 42 груши. В какой системе счисления посчитаны деревья?

- a. В шестеричной
- b. В шестнадцатеричной +
- c. В двоичной
- d. В восьмеричной

11 Умножьте в двоичной системе счисления $1111*11=$

- a. 1111111
- b. 101111
- c. 1010101
- d. 101101 +

13 Выполните умножение в системе счисления $10101*101=$

- a. 110101011
- b. 1101001
- c. 111001001+
- d. 1010101

15 В какой системе счисления будет верным равенство $7+8=16$?

- a. 12
- b. 6 +
- c. 9
- d. 16

16 $FFFF+1=$

- a. 1FFF +
- b. 10000
- c. FFFE
- d. FFF1

17 Выполните вычитание в двоичной системе счисления $1111-101$

- a. 1010
- b. 1110
- c. 1100
- d. 1001 +

18 Результат деления в двоичной системе счисления числа 110 на 11 равен

- a. 11
- b. 10
- c. 01
- d. 101 +

19 Перевести смешанное двоичное $111100000,101$ число в восьмеричную систему счисления

- a. 610,4
- b. 740,5
- c. 170,5
- d. 1E0,A +

20 Переведите восьмеричное число 0,25 в двоичную систему счисления

- a. 0,010101
- b. 0,101010 +

c. 0,110111

d. 0,101011

21 Алгоритм-это:

Указание на выполнение действий,

a. Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи, +

b. Процесс выполнения вычислений, приводящих к решению задачи

22 Свойство алгоритма – дискретность, выражает, что:

a. Команды должны следовать последовательно друг за другом, +

b. Каждая команда должна быть описана в расчете на конкретного исполнителя,

c. Разбиение алгоритма на конечное число команд

23 Формальное исполнение алгоритма – это:

a. Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,

b. Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,

c. Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем автоматически

+

d. Исполнение алгоритма осуществляется исполнителем на уровне его знаний

24 Самое важное свойство алгоритма:

a. Визуальность,

b. Массовость, +

c. Дискретность,

d. Аудиальность,

25 Какой алгоритм называется линейным:

a. Выполнение операций зависит от условия,

b. Операции выполняются друг за другом, +

c. Одни и те же операции выполняются многократно

d. Присутствие всех возможных операций в одном алгоритме

26 Графическое задание алгоритма – это:

a. Способ представления алгоритма с помощью геометрических фигур, +

b. Представление алгоритма в форме таблиц и расчетных формул,

c. Система обозначений и правил для единообразной и точной записи

алгоритмов и их исполнения

27 В расчете на кого должен строиться алгоритм:

a. В расчете на ЭВМ,

b. В расчете на умственные способности товарища,

c. В расчете на конкретного исполнителя +

28 Какое из перечисленных свойств относится к свойствам алгоритма:

a. Визуальность,

b. Совокупность,

c. Аудиальность,

d. Понятность +

29 Псевдокоманда DB означает

a. определить байт; +

b. определить слово (2 байта);

c. определить двойное слово (4 байта);

d. определить 6 байт

e. определить учетверенное слово (8 байт);

30 Псевдокоманда DW означает

a. определить байт;

b. определить слово (2 байта); +

c. определить двойное слово (4 байта);

d. определить 6 байт

e. определить учетверенное слово (8 байт);

Задачи, модельные ситуации или проблемы, ориентированные на профессиональную деятельность

1. Запросите реестр и выведите информацию о версии программного обеспечения браузера Internet Explorer или MS Edge, установленного на компьютере.

```
import win32api
import win32con
```

```
key = win32api.RegOpenKey(win32con.HKEY_LOCAL_MACHINE,
'SOFTWARE\\Microsoft\\Internet Explorer', 0,
win32con.KEY_ALL_ACCESS)
print(win32api.RegQueryValue(key, ''))
print(win32api.RegQueryValueEx(key, 'Version'))
print(win32api.RegQueryInfoKey(key))
win32api.RegCloseKey(key)
```

Результат:

```
('9.11.17134.0', 1)
(34, 10, 132302030397159009)
```

2. Проверьте список программ, которые запускаются при старте системы.

```
from win32api import *
from win32con import *
```

```
def get_values(fullname):
    name = str.split(fullname, '\\', 1)
    try:
        if name[0] == 'HKEY_LOCAL_MACHINE':
            key = RegOpenKey(HKEY_LOCAL_MACHINE, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CURRENT_USER':
            key = RegOpenKey(HKEY_CURRENT_USER, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CLASSES_ROOT':
            key = RegOpenKey(HKEY_CLASSES_ROOT, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_CURRENT_CONFIG':
            key = RegOpenKey(HKEY_CURRENT_CONFIG, name[1], 0, KEY_READ)
        elif name[0] == 'HKEY_USERS':
            key = RegOpenKey(HKEY_USERS, name[1], 0, KEY_READ)
        else:
            raise ValueError('Error, no key named', name[0])
        info = RegQueryInfoKey(key)
        for i in range(0, info[1]):
            ValueName = RegEnumValue(key, i)
            print(str.ljust(ValueName[0], 20), ValueName[1])
        RegCloseKey(key)
    except Exception as e:
        print('Sth is wrong')
        print(e)
```

```
if __name__ == '__main__':
```

```
    KeyNames = [
'HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run',
```

```
    r'HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunOnce',
    r'HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run',
    r'HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\RunOnce']
    for KeyName in KeyNames:
        print('*'*30)
        print(KeyName)
        get_values(KeyName)
```

Результат:

```

*****
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
SecurityHealth      %ProgramFiles%\Windows Defender\MSASCuiL.exe
Sysdiag             "C:\Program Files (x86)\Huorong\Sysdiag\bin\HipsTray.exe"
*****

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
*****

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
OneDrive            "C:\Users\ASUS\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
Wechat              C:\Program Files (x86)\Tencent\WeChat\WeChat.exe
QQ2009              "E:\QQbrower\Bin\QQ.exe" /background
BaiduYunGuanjia     "E:\云盘\BaiduNetdisk\baidunetdisk.exe" AutoRun
BaiduYunDetect      "E:\云盘\BaiduNetdisk\YunDetectService.exe"
qbclipboard         "C:\Program Files (x86)\Tencent\QQBrowser\QQBrowser.exe" --type=assistant --clipboard
AlibabaMusic        C:\Program Files (x86)\Xiami\XiamiPC.exe --tray
Thunder             E:\Thunder\Program\Thunder.exe -silent -StartType:AutoRun
*****

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce

```

3. Перечислите значения реестра ветки

Software\Microsoft\Windows\CurrentVersion\Explorer».

```

import winreg

key = winreg.OpenKey(winreg.HKEY_CURRENT_USER,
r'Software\Microsoft\Windows\CurrentVersion\Explorer')
try:
    i = 0
    while 1:
        Name, Value, Type = winreg.EnumValue(key, i)
        print(repr(Name), ': ', repr(Value), ': ', Type)
        i += 1
except WindowsError as e:
    # print(e)
    pass

"""
_winreg.QueryValueEx(key,value_name)
Эта функция используется для получения данных и типа указанного значения в ключе, а
тип возвращаемого значения - кортеж.
"""
print('=' * 20)
Name = 'link'

Value, Type = winreg.QueryValueEx(key, Name)
print(Name, Value)

```

Результат:

[illegible]

```

Код сервера
import socket

words = {'how are you?': 'Fine,thank you', 'how old are you': '3', 'what is your name?':
'TianMaoSpirit',
        'where do you work': 'CQUPT', 'bye': 'Bye'} # Когда вы узнаете о разработке базы
данных позже, вы можете использовать поисковые роботы для получения и сохранения
данных в базе данных, а затем получить большое количество словарей из базы данных
HOST = "
PORT = 50007
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Привязать сокет
s.bind((HOST, PORT))
# Начать слушать
s.listen(1)
print('Listening at port:', PORT)
client_socket, addr = s.accept() # Пассивно принимать TCP-клиентские соединения,
(блокируя) ожидая прибытия соединения, если клиентский запрос получен, сокет и адрес
клиентского сокета будут возвращены
print('Connected by', addr)
while True:
    data = client_socket.recv(1024)
    data = data.decode()
    if not data:

```

```

        break
    print('Received message:', data)
    client_socket.sendall(words.get(data, 'nothing').encode())
client_socket.close()
s.close()

```

Код клиента

```

import socket
import sys

HOST = '127.0.0.1' # IP-адрес хоста сервера
PORT=50007 # Номер порта хоста сервера
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
try:
    s.connect((HOST,PORT))
except Exception as e:
    print('Server not found or not open')
    sys.exit()
while True:
    c = input('Input the content you want to send:')
    s.sendall(c.encode()) # отправить данные
    data = s.recv(1024) # Получать данные с сервера, установить размер кеша на 1 м = 1024
    бит
    data = data.decode()
    print('Received:',data)
    if c.lower()=='bye':
        break
    s.close() # Закрывать соединение

```

Результат:

```

Received message: how are you?
Received message: where do you work?
Received message: how old are you
Received message: what is your name?
Received message: bye
Input the content you want to send:how are you?
Received: Fine,thank you
Input the content you want to send:where do you work?
Received: nothing
Input the content you want to send:how old are you
Received: 3
Input the content you want to send:what is your name?

```

**Оценочные материалы текущего контроля знаний и промежуточной аттестации по
МДК 02.02 Инструментальные средства разработки программного
обеспечения**

Компетенция ПК 2.2

1. Какие технологии разработки программ используются в современном программировании: 1. Визуальные 2. Событийные 3. Структурные 4. Объектно-ориентированные 5. Модульные 6. Текстуальные 7. Графические 8. Машинно-ориентированное
2. Инструментальные средства разработки программ – это ... 1. Средства создания новых программ 2. Сервисные средства разработки ПО 3. Аналитические средства разработки ПО 4. Программное обеспечение, предназначенное для разработки и отладки новых программ 5. Средства отладки ПО 6. Средства тестирования ПО 7. Аппаратные и программные инструменты разработки нового ПО Технические инструментальные средства разработки ПО
3. Укажите правильную последовательность создания программного обеспечения: 1. Анализ требований, проектирование, программирование, тестирование 2. Формулирование задачи, анализ требований, программирование, проектирование, тестирование 3. Анализ требований, программирование, проектирование, отладка, тестирование 4. Формулирование задачи, анализ требований, проектирование, программирование, тестирование
4. В чем сущность модульного программирования: 1. в разбиении программы на отдельные функционально независимые части 2. в разбиении программы на отдельные равные части 3. в разбиение программы на процедуры и функции
5. Недостаток модульного программирования: 1. увеличивает трудоемкость программирования; 2. снижает быстродействие программы; 3. не позволяет выполнять оптимизацию программы. 4. усложняет процедуру комплексного тестирования
6. При структурном программировании задача выполняется: 1. поэтапным разбиением на более легкие задачи; 2. без участия программиста; 3. объединением отдельных модулей программы.
7. Модульное программирование применимо при: 1. проектировании сверху-вниз; 2. проектирование снизу-вверх;
8. Проектирование сверху-вниз это: 1. последовательное разбиение общих задач на более мелкие; 2. составление из отдельных модулей большой программы.
9. Проектирование снизу-вверх это: 1. составление из отдельных модулей большой программы; 2. последовательное разбиение общих задач на более мелкие.
10. Достоинство модульного программирования: 1. не требует комплексного тестирования 2. уменьшает размер программы 3. возможность приступить к тестированию до завершения написания всей программы 4. повышает надежность программы

11. Рекомендуемые размеры модулей: 1. большие; 2. равные; 3. небольшие; 4. фиксированной длины.
12. В чем заключается независимость модуля: 1. в написании, отладке и тестировании независимо от остальных модулей; 2. в разработке и написании независимо от других модулей; 3. в независимости от работы основной программы.
13. Основные характеристики модуля 1. Размер 2. Прочность 3. Сцепление 4. Рутинность 5. Связность 6. Безопасность 7. Определенность
14. К основным понятиям объектно-ориентированного подхода относятся: 1. Обобщение 2. Полиморфизм 3. Инкапсуляция 4. Реализация 5. Агрегирование 6. Наследование 7. Ассоциация 8. Композиция
15. Какие методы сборки программы существуют 1. монолитная 2. пошаговая 3. одновременная 4. постепенная
16. Программное средство, обеспечивающее автоматизированную поддержку деятельности, выполняемой в рамках технологических операций. 1. Инструментальное средство (CASE-средство) 2. Операционная система 3. Текстовый редактор 4. Язык программирования
17. Что такое управление версиями: 1. автоматизированный процесс трансформации исходных текстов ПО в пакет исполняемых модулей 2. управление версиями файлов 3. ручной процесс трансформации исходных текстов ПО в пакет исполняемых модулей
18. Требования к информационной системе сформулированы в 1. описании предметной области 2. техническом проекте 3. техническом задании 4. продукте моделирования
19. Целью разработки архитектуры будущего программного обеспечения является 1. разработка устройств основных компонент программного обеспечения. 2. разработка программного кода 3. тестирование 4. разработка модели (описания) будущей системы, понятной для кодировщика
20. Требования к информационной системе реализованы в 1. описании предметной области 2. техническом проекте

3. техническом задании
4. продукте моделирования
21. Функциональные характеристики применения, являющиеся исходными данными для оценки и выбора технических средств для разработки программного обеспечения 1. среда функционирования 2. удобство использования 3. совместимость с другими ТС ПО 4. соответствие технологическим стандартам
22. К числу основных возможностей, обеспечиваемых современными инструментальными средствами, относятся: 1. графический анализ и проектирование; 2. интерактивное прототипирование; 3. автоматическое тестирование и верификация программного обеспечения; 4. разработка руководства пользователей.
23. Достоинство модульного программирования: 1. не требует компоновки 2. создание программы по частям в произвольном порядке 3. всегда дает эффективные программы 4. снижает количество ошибок
24. Что не входит в категорию «Инструментальные средства разработки программного обеспечения»: 1. средства разработки приложений клиент-сервер; 2. средства разработки Internet/Intranet-приложений; 3. СУБД (системы управления базами данных); 4. средства автоматизации проектирования программного обеспечения.
25. Что не относится к системному программному обеспечению ЭВМ включает: 1. операционные системы; 2. системы автоматизированного проектирования; 3. утилиты; 4. драйвера; 5. операционные оболочки.
26. Программное обеспечение ЭВМ можно классифицировать на следующие группы: 1. системное программное обеспечение; 2. прикладное программное обеспечение; 3. операционные системы; 4. пакеты прикладных программ общего назначения; 5. инструментальные средства разработки программного обеспечения.
27. Прикладное программное обеспечение включает: 1. системы автоматизированного проектирования; 2. утилиты; 3. драйвера; 4. текстовые редакторы; 5. программы автоматизации бухгалтерского учёта.
28. Модуль – это отдельная, функционально законченная программная единица, которая 1. может быть частью программы, но не может применяться самостоятельно 2. не может применяться самостоятельно, либо быть частью программы 3. может применяться самостоятельно, но не может быть частью программы 4. может применяться самостоятельно, либо быть частью программы
29. Стил программирования - это 1. собрание правил, которых обязан придерживаться каждый программист при написании приложений 2. четко определенный кодекс «статей» программирования на наиболее распространенных языках (C++, Java и т.п.) 3. набор приемов и методов программирования, позволяющих создавать корректные, эффективные и доступные для чтения и понимания программы

4. выбор определенного языка программирования разработчиком при постановке перед ним определенной задачи

30. Интеграция программных продуктов с другими программами называется 1. коммуникативностью 2. мобильностью 3. эффективностью 4. модифицируемостью
31. Программное обеспечение включает: 1. Исходные тексты программ, исполнимый код 2. Техническое задание, исходные тексты программ 3. Исполнимый код, программную документацию
32. Репозиторий-портал 1. совмещает в себе функции каталога ПО, инструмента для организации сообществ по разработке ПО, среды информационного обмена для разработчиков, технической инфраструктуры разработки проектов по созданию ПО и инструмента для загрузки разработанного ПО. 2. направлен на конечных пользователей ПО, который может выступать как хранилищем файлов, так и хранилищем ссылок на другие сайты, где можно загрузить соответствующие файлы. 3. служит для интеграции разных пакетов программ в единую систему.
33. Репозиторий-каталог 1. совмещает в себе функции каталога ПО, инструмента для организации сообществ по разработке ПО, среды информационного обмена для разработчиков, технической инфраструктуры разработки проектов по созданию ПО и инструмента для загрузки разработанного ПО. 2. направлен на конечных пользователей ПО, который может выступать как хранилищем файлов, так и хранилищем ссылок на другие сайты, где можно загрузить соответствующие файлы. 3. служит для интеграции разных пакетов программ в единую систему.
34. Что делает команда git add? 1. Создает файл с указанным именем и сразу добавляет его в Git 2. Добавляет локальный файл в удаленный репозиторий так, чтобы другие участники проекта могли его видеть 3. Начинает отслеживать указанный файл или файлы
35. Что делает команда git status? 1. Показывает состояние проекта 2. Показывает имя и email нашего пользователя 3. Показывает место, занимаемое репозиторием на жестком диске и количество выделенного под репозиторий месте 4. Такой команды нет, есть только команда git show
36. Что такое Git Hub? 1. Программа для работы с Git 2. Драйвер для Git 3. Веб-сервис для хостинга IT-проектов и их совместной разработки, основанный на Git 4. UI для работы с локальной версией Git
37. Что такое ветка в репозитории Git? 1. Это то же самое, что и коммит 2. Это минимум два коммита с одинаковым коммит-сообщением 3. Это разные пути развития проекта, по сути разные последовательности коммитов 4. Это механизм изменения конкретного файла
38. Что такое коммит? 1. Это единица состояния проекта в Git 2. Это результат вывода команды git diff 3. Это обобщающее название одного из статусов файла в выводе git status: untracked, new, deleted или modified 4. Это слово ничего не означает, его ввели только для того, чтобы путать новичков

39. Что такое репозиторий Git? 1. Любая директория/папка в моей ОС 2. Любая папка, находящаяся внутри Git 3. Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содержащее собственно файлы 4. Папка .git/ и все входящие в нее
40. Головной модуль программной системы 1. управляет запуском программного продукта (существует в единственном числе); 2. обеспечивает вызов других модулей на обработку; 3. выполняют функции обработки; 4. осуществляют обслуживающие функции.
41. Управляющий модуль программной системы 1. управляет запуском программного продукта (существует в единственном числе); 2. обеспечивает вызов других модулей на обработку; 3. выполняют функции обработки; 4. осуществляют обслуживающие функции.
42. Протокол UDP 1. протокол не требующий установки соединения, не гарантирует доставку своих сообщений 2. протокол требующий установки соединения, обеспечивает надежную транспортировку данных
43. Системы контроля версий делятся на: 1. локальные 2. централизованные 3. распределенные 4. примитивные
44. В какой ситуации надо делать git status? 1. Чем чаще, тем лучше 2. Всегда при создании коммита 3. Всегда после команды git pull 4. Только если надо узнать, в каком статусе находится репозиторий, а так эта команда не является обязательной для любой манипуляции
45. Протокол TCP 1. протокол не требующий установки соединения, не гарантирует доставку своих сообщений 2. протокол требующий установки соединения, обеспечивает надежную транспортировку данных
46. Какие задачи выполняют уровни OSI в процессе передачи данных по сети: 1. уровни выполняют одинаковые задачи, постоянно повторяя передающие сигналы по сети 2. каждый уровень выполняет свою определенную задачу 3. первых три уровня выполняют одинаковые задачи, последующие выполняют определенные задачи
47. Протокол управления передачей данных – это 1. IP 2. TCP 3. FTP 4. SMTP
48. Протокол – это 1. Специальная программа, передающая данные по сети. 2. Набор соглашений, регулирующих передачу данных по сети. 3. Специальное устройство компьютера, управляющее передачей данных. 4. Специальная область жесткого диска, через которую производится передача данных.
49. Протокол TCP/IP относится 1. К аппаратному уровню сети Internet.

2. К системному (сетевому или транспортному) уровню сети Internet. 3. К сеансовому уровню сети Internet. 4. К прикладному уровню сети Internet.
50. Процедуру условного форматирования можно применять только: 1. только к числовым значениям 2. только к текстовым данным 3. только к ячейкам, содержащим формулу либо функцию 4. ко всем перечисленным значениям

Компетенция ПК 2.3

1. Отладка ПС - это ... 1. деятельность, направленная на обнаружение и исправление ошибок в ПС с использованием процессов выполнения его программ 2. процесс выполнения программы на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этой программы 3. построение программы как композиции из нескольких типов управляющих конструкций, которые позволяют повысить понимаемость логики работы программы
2. Отладчик: 1. Программа для создания системы защиты файла 2. Программа создания системы защиты от вирусных атак 3. Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку 4. Операционная оболочка для создания и управления файловыми структурами 5. Системное программное обеспечение для настройки операционной системы 6. Программа создания и редактирования графических файлов 7. Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения
3. Процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. 1. Отладка 2. Локализация 3. Тестирование
4. На каком этапе создания программы могут появиться синтаксические ошибки 1. Программирование 2. Проектирование 3. Тестирование 4. Анализ требований
5. Отладка программ это: 1. локализация и исправление ошибок 2. алгоритмизация программирования 3. компиляция и компоновка
6. Вид ошибки с неправильным написанием служебных слов (операторов) 1. синтаксическая 2. семантическая 3. логическая 4. символьная
7. Вид ошибки с неправильным использованием служебных слов (операторов) 1. синтаксическая 2. семантическая 3. логическая 4. символьная
8. Причины синтаксических ошибок: 1. ошибки в исходных данных; 2. ошибки, допущенные на более ранних этапах; 3. плохое знание языка программирования; 4. неправильное применение процедуры тестирования.

9. Комплекс программных средств, предназначенных для кодирования, тестирования и отладки программного обеспечения. Имеет интерфейс, удобный пользователю. 1. Система программирования 2. Компилятор 3. Синтаксический анализатор 4. Средства автоматизации сборки
10. Отладчик: 1. Программа, облегчающая программисту выполнение отладки разрабатываемых им программ 2. Программа для создания системы защиты файла 3. Программа создания системы защиты от вирусных атак 4. Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку 5. Операционная оболочка для создания и управления файловыми структурами 6. Системное программное обеспечение для настройки операционной системы 7. Программа создания и редактирования графических файлов 8. Программа, позволяющая выполнять остановы в заданных точках, просмотреть текущие значения переменных и изменять их значения
11. Последовательность этапов программирования 1. компилирование, компоновка, отладка 2. компоновка, отладка, компилирование 3. отладка, компоновка, компилирование 4. компилирование, отладка, компоновка
12. Вставьте пропущенное слово. _____, приложения, выполняющие программу в заданном режиме (например, пошаговом) с целью поиска, обнаружения и локализации ошибок. Используются на этапе компиляции. Ответ: отладчики
13. Выберите базовые функции отладчика: 1. ввод и модификация текст программы 2. пошаговое выполнение программы (режим трассировки) с отображением результатов 3. остановка в заранее определенных точках 4. перевод исходного текста на входном языке в язык машинных кодов 5. возможность остановки в некотором месте программы при выполнении некоторого условия 6. изображение и изменение значений переменных 7. объединяют несколько объектных модулей, порождаемых компилятором, в одну программ
14. Логические ошибки 1. Это ошибки, обнаруженные компилятором. Их можно подразделить на категории в зависимости от того, какие правила языка он нарушают 2. Это ошибки, обнаруженные в ходе контрольных проверок выполняемого модуля. 3. Это ошибки, найденные программистом в поисках причины неправильных результатов. 4. Это ошибки, обнаруженные редактором связей при попытке объединить объектные файлы в выполняемый модуль
15. Инструментальные средства отладки (Не правильный ответ) 1. отладчики 2. трассировка 3. компиляторы
16. Когда программист может проследить последовательность выполнения команд программы: 1. при тестировании 2. при трассировке 3. при компиляции 4. при выполнении программы

5. при компоновке
17. Синтаксические ошибки - ошибки, фиксируемые компилятором (транслятором, интерпретатором) при выполнении синтаксического и частично семантического анализа программы; - ошибки, обнаруженные компоновщиком (редактором связей) при объединении модулей программы; - ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы.
18. Ошибки компоновки - - ошибки, фиксируемые компилятором (транслятором, интерпретатором) при выполнении синтаксического и частично семантического анализа программы; ошибки, обнаруженные компоновщиком (редактором связей) при объединении модулей программы; - ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы.
19. Ошибки выполнения - ошибки, фиксируемые компилятором (транслятором, интерпретатором) при выполнении синтаксического и частично семантического анализа программы; ошибки, обнаруженные операционной системой, аппаратными средствами или пользователем при выполнении программы. - ошибки, обнаруженные компоновщиком (редактором связей) при объединении модулей программы
20. Попытка найти ошибки при выполнении программы в реальной среде: - отладка - контроль испытание
21. Процесс локализации и исправления ошибок, обнаруженных при тестировании ПО называют: отладкой - локализацией - инициализацией
22. Что относится к ошибкам кодирования: - ошибки выполнения ошибки некорректного использования переменных, ошибки вычислений, ошибки взаимодействия модулей, игнорирование особенностей конкретного языка программирования - логические ошибки
23. Какой метод отладки программ описан в тексте: «Самый простой и естественный способ отладки программы. Метод эффективен, но не применим для программ со сложными вычислениями, для больших программ, а также в случаях, когда ошибка связана с неверным представлением программиста о выполнении операций»: - метод индукции метод ручного тестирования - метод обратного прослеживания
24. Какой метод отладки программ описан в тексте: «Сначала формируют множество причин, которые могли бы вызвать данное проявление ошибки. Затем, анализируя причины, исключают те, которые противоречат имеющим данным.»: - метод индукции - метод ручного тестирования метод дедукции
25. Какой метод отладки программ описан в тексте: «Метод основан на тщательном анализе симптомов ошибки, которые могут проявляться как неверные результаты вычислений или как сообщение об ошибке.» метод индукции - метод ручного тестирования

3. метод дедукции
26. Какой метод отладки программ описан в тексте: «Начинается проверка с точки вывода неправильного результата. Для этой точки строится гипотеза о значениях основных переменных, которые могли бы привести к получению имеющегося результата» 1. метод индукции 2. метод ручного тестирования 3. метод обратного прослеживания
27. Последовательность этапов отладки 1. локализация ошибок, определение причины ошибки, исправление ошибки, изучение проявление ошибки 2. Изучение проявление ошибки, определение причины ошибки, локализация ошибок, исправление ошибки 3. Изучение проявление ошибки, локализация ошибок, повторное тестирование, определение причины ошибки, исправление ошибки 4. изучение проявление ошибки, локализация ошибок, определение причины ошибки, исправление ошибки, повторное тестирование
28. Сложный алгоритм пишется или прототипируется на одном языке программирования (с наличием всех доступных инструментальных средств), а потом исходный код отлаженного алгоритма транслируется вручную или автоматически в другой язык программирования (целевой системы), для которого отсутствуют необходимые инструментальный средства 1. Отладка методом RPC 2. Отладка трансляцией кода 3. Профилирование кода 4. Отладка разработкой интерпретатора
29. Логирование кода 1. выполнение автоматических unit test-ов в более простых сценариях для функций и автоматическое выявление проблемных участков кода 2. вывод в файл входных, выходных аргументов функций, промежуточных состояний в процессе исполнения программы 3. поиск причин возникновения дефекта с помощью анализа исходного кода программы, проблемного контента, конфигурации, состояния базы данных и т.п. 4. изолирование проблемы используя ручное или автоматическое тестирование
30. Динамические анализаторы 1. контролируют время выполнения участков кода программы, находят точки (области) некорректной работы с памятью и объектами операционной системы и др. 2. выявляют ошибки выхода (обращения) за границы массивов, потенциальные проблемы безопасности, т. н. утечки памяти, некорректность использования системных ресурсов и др.
31. Статические анализаторы 1. контролируют время выполнения участков кода программы, находят точки (области) некорректной работы с памятью и объектами операционной системы и др. 2. выявляют ошибки выхода (обращения) за границы массивов, потенциальные проблемы безопасности, т. н. утечки памяти, некорректность использования системных ресурсов и др.
32. Дефект 1. участок программного кода, выполнение которого при определенных условиях приводит к неожиданному поведению системы 2. кратковременная самоустраняющаяся утрата работоспособности технического устройства 3. это проявление ошибки в программной системе и выражается в утрате работоспособности
33. Отказ 1. участок программного кода, выполнение которого при определенных условиях приводит к неожиданному поведению системы 2. проявление ошибки в программной системе и выражается в утрате работоспособности 3. кратковременная самоустраняющаяся утрата работоспособности технического устройства

34. Сбой 1. участок программного кода, выполнение которого при определенных условиях приводит к неожиданному поведению системы 2. кратковременная самоустраняющаяся утрата работоспособности технического устройства 3. проявление ошибки в программной системе и выражается в утрате работоспособности
35. Верификация 1. процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа 2. определение соответствия, разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе 3. процесс изменения программной системы таким образом, что её внешнее поведение не изменяется, а внутренняя структура улучшается
36. Валидация 1. процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа 2. определение соответствия, разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе 3. процесс изменения программной системы таким образом, что её внешнее поведение не изменяется, а внутренняя структура улучшается
37. Рефакторинг 1. процесс оценки системы или её компонентов с целью определения удовлетворяют ли результаты текущего этапа разработки условиям, сформированным в начале этого этапа. 2. определение соответствия, разрабатываемого ПО ожиданиям и потребностям пользователя, требованиям к системе 3. процесс изменения программной системы таким образом, что её внешнее поведение не изменяется, а внутренняя структура улучшается
38. Тестирование, предназначенное для проверки правильности функционирования методов классов ПО 1. Модульное тестирование 2. Исследовательское тестирование 3. Функциональное тестирование 4. Нагрузочное тестирование 5. Регрессионное тестирование
39. Тестирование, предназначенное для проверки работоспособности программного продукта при предельной входной нагрузке 1. Модульное тестирование 2. Исследовательское тестирование 3. Функциональное тестирование 4. Нагрузочное тестирование 5. Регрессионное тестирование
40. Тестирование, которое применяется при внесении изменений в программное обеспечение с целью проверки корректности работы компонентов системы, которые потенциально могут взаимодействовать с измененным компонентом 1. Модульное тестирование 2. Исследовательское тестирование 3. Функциональное тестирование 4. Нагрузочное тестирование 5. Регрессионное тестирование
41. Обработчик событий - это... 1. метод программирования, основанным на управлении событиями 2. визуальное отображение окна приложения 3. процедура (или функция) инициируемая событием 4. совокупность данных и методов работы с ними

42. Исключительные ситуации делятся 1. Контролируемые 2. Исправимые 3. Неисправимые
43. Исключения делятся на 1. Аппаратные 2. Пользовательские 3. Программные
44. ... - это отклонение программы от функционирования или невозможность программы выполнять функции, определенные требованиями и ограничениями, что рассматривается как событие, способствующее переходу программы в неработоспособное состояние из-за ошибок, скрытых в ней дефектов или сбоев в среде функционирования. 1. Ошибка (error) 2. Отказ (failure) 3. Дефект (fault)
45. ... - состояние программы, при котором выдаются неправильные результаты, причиной которых являются изъяны в операторах программы или в технологическом процессе ее разработки, что приводит к неправильной интерпретации исходной информации, следовательно, и к неверному решению. 1. Ошибка (error) 2. Отказ (failure) 3. Дефект (fault)
46. ... - следствие ошибок разработчика на любом из этапов разработки, которая может содержаться в исходных или проектных спецификациях, текстах кодов программ, эксплуатационной документации и т.п. 1. Ошибка (error) 2. Отказ (failure) 3. Дефект (fault)
47. Способы оценки надежности 1. Тестирование 2. Сравнение с аналогами 3. Трассировка 4. Оптимизация
48. Когда можно обнаружить синтаксические ошибки 1. при отладке 2. при тестировании 3. при компиляции 4. при эксплуатации 5. на этапе проектирования
49. Ошибки компоновки заключается в том, что 1. Указано внешнее имя, но не объявлено 2. Составлено неверное выражение 3. Неправильно использовано зарезервированное слово 4. Указан неверный тип переменной
50. Программа для просмотра значений переменных при выполнении программы: 1. Отладчик 2. Компилятор 3. Интерпретатор

Компетенция ПК 2.5

1. Отладка программ это: 1. локализация и исправление ошибок 2. алгоритмизация программирования 3. компиляция и компоновка
2. Компилятор – это:

1. Прикладное программное обеспечение 2. Специальная утилита системного ПО 3. Операционная оболочка 4. Программное обеспечение, используемое в издательских системах 5. Программа, которая переводит программу, написанную на языке программирования высокого уровня в программу на машинном языке
3. Отладчик: 1. Программа для создания системы защиты файла 2. Программа создания системы защиты от вирусных атак 3. Программа, помогающая анализировать поведение отлаживаемой программы, обеспечивая ее трассировку 4. Операционная оболочка для создания и управления файловыми структурами 5. Системное программное обеспечение для настройки операционной системы 6. Программа создания и редактирования графических файлов 7. Программа, позволяющая выполнять остановки в заданных точках, просмотреть текущие значения переменных и изменять их значения
4. Процесс локализации и исправления ошибок, обнаруженных при тестировании программного обеспечения. 1. Отладка 2. Локализация 3. Тестирование
5. Определение оператора/операторов программы, выполнение которого вызвало нарушение вычислительного процесса. 1. Отладка 2. Локализация 3. Тестирование
6. Когда программист может проследить последовательность выполнения команд программы: 1. при тестировании 2. при трассировке 3. при компиляции 4. при выполнении программы 5. при компоновке
7. Процесс исполнения программы с целью обнаружения ошибок: 1. кодирование 2. сопровождение 3. тестирование 4. проектирование
8. На языке программирования составляется: 1. исходный код 2. исполняемый код 3. объектный код 4. алгоритм
9. Одно из преимуществ автоматизации программирования: 1. наглядное программирование с визуальным контролем 2. получение стандартной программы 3. создание программы с оптимальным кодом
10. В чем отличия формальной инспекции от тестирования? 1. не происходит выполнения программного кода 2. может применяться как к требованиям, так и к коду 3. более быстрый процесс 4. не фиксируются найденные проблемы
11. На каком этапе создания программы могут появиться синтаксические ошибки 1. Программирование 2. Проектирование

3. Тестирование
4. Анализ требований
12. Верификация это 1. процесс проверки соответствия поведения системы требованиям 2. процесс устранения ошибок в программном обеспечении 3. процесс взаимодействия с пользователем, направленный на улучшение его понимания принципов работы программной системы 4. процесс уточнения требований по результатам обсуждения с пользователем
13. Вид ошибки с неправильным использованием служебных слов (операторов) 1. синтаксическая 2. семантическая 3. логическая 4. символьная