

**ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ
ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ
ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ**

по специальности 09.02.07 Информационные системы и программирование
квалификация Программист

Комплект ФОС ГИА включен в учебно-методическую документацию для аттестации обучающихся на соответствие требованиям образовательной программы по специальности 09.02.07 Информационные системы и программирование. Рассмотрено и одобрено на заседании отделения
Разработчик: НПОЧУ «Колледж экономики и права»

СОГЛАСОВАНО

Заместитель директора по УВР

09 января 2024 г.

СОДЕРЖАНИЕ

	Стр.
1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ	4
2 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ (РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ)	16
3 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	55
4 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ПО УЧЕБНОЙ И (ИЛИ) ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ	61

1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ ПО ПМ.01. РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

1.1 Общие положения

Формирование фондов оценочных средств (далее - ФОС) - необходимое условие реализации основной профессиональной образовательной программы. Под фондом оценочных средств понимается комплект методических и контрольно-измерительных материалов, предназначенных для оценивания знаний, умений, сформированности общих и профессиональных компетенций на разных стадиях обучения. ФОС по учебной дисциплине ПМ.01

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем состоит из КИМ (контрольно-измерительные материалы) для текущего и рубежного контроля знаний и умений, обучающихся и КОС (контрольно-оценочные средства) для проведения промежуточной аттестации в соответствии с учебным планом.

Контрольно-измерительные материалы и контрольно-оценочные средства разработаны на основе ФГОС СПО по специальности 09.02.07 Информационные системы и программирование и предназначены для оценки образовательных достижений обучающихся. КИМ и КОС позволяют оценить знания, умения, сформированность общих и профессиональных компетенций, обучающихся на соответствие (или несоответствие) уровня их подготовки требованиям ФГОС СПО по освоению ПМ.01. Разработка модулей программного обеспечения для компьютерных систем.

Формы промежуточного контроля по МДК являются:

- МДК 01.01. Разработка программных модулей - экзамен в 7 семестре
- МДК 01.02 Поддержка и тестирование программных модулей - дифференцированный зачет в 7 семестре
- МДК 01.03. Разработка мобильных приложений - экзамен в 7 семестре
- МДК 01.04. Системное программирование - экзамен в 7 семестре
- Учебная практика - дифференцированный зачет в 6 семестре
- Производственная практика - дифференцированный зачет в 7 семестре
- ПМ.01 - квалификационный экзамен в 7 семестре
 - формам текущего контроля по ПМ.01. Разработка модулей программного обеспечения для компьютерных систем относятся:
- устный опрос по теме;
 - подготовка сообщений и докладов;
 - подготовка презентаций;
 - компьютерное тестирование по отдельным темам;
 - отчет по практической работе;
 - выполнение аудиторной самостоятельной работы.

Разработка оценочных материалов для включения в КОС проводилась с учетом:

- форм проведения оценочных мероприятий (устный опрос, самостоятельная аудиторная работа, тестирование, в т.ч. компьютерное);
- уровней освоения учебного материала темы (ознакомительный, репродуктивный, продуктивный);
- видов деятельности, которые будут выполнять обучающиеся в процессе оценочных мероприятий (осознанное воспроизведение информации, применение информации, анализ, синтез, оценка);
- обучающих возможностей оценочных материалов;
 - возможности принятия решения об освоении обучающимися общих и профессиональных компетенций.

В состав КОС включены материалы, выполняющие как контролирующие, так и обучающие функции. Они позволяют не только проверить уровень усвоения знаний, освоения умений, но и оценить различные качества личности обучающегося, уровень сформированности профессиональных и общих компетенций.

Чтобы обеспечить объективную оценку результатов контроля, преподавателем разработаны критерии оценки показателей результатов обучения, эталоны выполнения заданий, «ключи» к тестам и т.п.

Тесты (в т.ч. для проведения компьютерного тестирования) формируются в соответствии с общими требованиями к оформлению и содержанию тестов.

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем является обязательной частью профессионального цикла программы подготовки специалистов среднего звена в соответствии с ФГОС по специальности 09.02.07 Информационные системы и программирование.

ПМ.01. Разработка модулей программного обеспечения для компьютерных систем направлен на формирование общих и профессиональных компетенций.

Код	Наименование видов деятельности и профессиональных компетенций
ВПД 01	<i>Разработка модулей программного обеспечения для компьютерных систем</i>
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
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 4	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 5	Планировать и реализовывать собственное профессиональное и личностное развитие.
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языках.

В результате освоения профессионального модуля обучающийся **должен:**

Иметь практический опыт	В разработке кода программного продукта на основе готовой спецификации на уровне модуля; использовании инструментальных средств на этапе отладки программного продукта; проведении тестирования программного модуля по определенному сценарию; использовании инструментальных средств на этапе отладки программного продукта; разработке мобильных приложений
уметь	осуществлять разработку кода программного модуля на языках низкого и высокого уровней; создавать программу по разработанному алгоритму как отдельный модуль; выполнять отладку и тестирование программы на уровне модуля; осуществлять разработку кода программного модуля на современных языках программирования; уметь выполнять оптимизацию и рефакторинг программного кода; оформлять документацию на программные средства
знать	основные этапы разработки программного обеспечения; основные принципы технологии структурного и объектно-ориентированного программирования; способы оптимизации и приемы рефакторинга; основные принципы отладки и тестирования программных продуктов

Перечень оценочных средств

1. Текущий контроль

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1	2	3	4
1.	Практические работы		Методическая разработка практических работ
2.	Тест по теме, разделу	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий
3.	Реферат, доклад, сообщение	Продукт самостоятельной работы студента, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы	Темы рефератов, докладов, сообщений
4.	Вопросы для устного (письменного) опроса по теме, разделу	Средство контроля, организованное как специальная беседа преподавателя со студентами на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины

2. Промежуточная аттестация

3.

Формы контроля	Контрольно-оценочная документация
Экзамен по ПМ.01	Журнал учебных занятий, экзаменационная ведомость

Типовые задания для оценки освоения модуля

Тема	Наименование	Контролируемые компетенции	Наименование оценочного средства
МДК 01.01. Разработка программных модулей			
Тема 1.2. Основные этапы разработки программного обеспечения	Практическая работа № 1.1. Формирование алгоритмов разработки программных модулей в соответствии с техническим заданием Практическая работа № 1.2. Оформление документации на программные средства	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 1.3. Методы программирования	Практическая работа № 1.3. Создание программ по разработанному алгоритму как отдельный модуль Практическая работа № 1.4. Разработка алгоритма поставленной задачи Практическая работа № 1.5. Реализация алгоритма поставленной задачи средствами автоматизированного проектирования Практическая работа № 1.6. Использование инструментальных средств на этапе отладки программного модуля	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 1.4. Структурное программирование	Практическая работа № 1.7. Оценка сложности алгоритмов сортировки Практическая работа № 1.8. Оценка сложности алгоритмов поиска Практическая работа № 1.9. Оценка сложности рекурсивных алгоритмов Практическая работа № 1.10. Оценка сложности эвристических алгоритмов	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 1.5. Объектно-ориентированное программирование	Практическая работа № 1.11. Работа с классами Практическая работа № 1.12. Перегрузка методов Практическая работа № 1.13. Определение операций в классе Практическая работа № 1.14. Создание наследованных классов Практическая работа № 1.15. Работа с объектами через интерфейсы Практическая работа № 1.16. Использование стандартных интерфейсов Практическая работа № 1.17. Работа с типом данных структура Практическая работа № 1.18. Коллекции Практическая работа № 1.19. Параметризованные классы Практическая работа № 1.20. Использование регулярных выражений Практическая работа № 1.21. Операции со списками	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)

Тема 1.6 . Паттерны проектирования	Практическая работа № 1.22. Использование основных шаблонов Практическая работа № 1.23. Использование порождающих шаблонов Практическая работа № 1.24. Использование структурных шаблонов Практическая работа № 1.25. Использование поведенческих шаблонов	ПК 1- ПК 5 ОК1-ОК10	Практическа я работа (отчет по практическо й работе)
Тема 1.7. Событийно - управляемое программирование	Практическая работа № 1.26. Разработка приложения с использованием текстовых компонентов Практическая работа № 1.27. Разработка приложения с несколькими формами Практическая работа № 1.28. Разработка приложения с не визуальными компонентами Практическая работа № 1.29. Разработка игрового приложения Практическая работа № 1.30. Разработка игрового приложения Практическая работа № 1.31. Разработка приложения с анимацией	ПК 1- ПК 5 ОК1-ОК10	Практическа я работа (отчет по практическо й работе)
Тема 1.8. Оптимизация и рефакторинг кода	Практическая работа № 1.32. Оптимизация кода Практическая работа № 1.33. Рефакторинг кода	ПК 1- ПК 5 ОК1-ОК10	Практическа я работа (отчет по практическо й работе)
Тема 1.9. Разработка пользовательского интерфейса	Практическая работа № 1.34. Пользовательская и программная модели интерфейса Практическая работа № 1.35. Разработка технического задания Практическая работа № 36. Разработка интерфейса пользователя Практическая работа № 1.37. Проектирование пользовательского интерфейса десктопного приложения Практическая работа № 1.38. Проектирование пользовательского интерфейса десктопного приложения Практическая работа № 1.39. Проектирование пользовательского интерфейса мобильного приложения Практическая работа № 1.40. Проектирование пользовательского интерфейса мобильного приложения Практическая работа № 1.41. Адаптивный веб-дизайн Практическая работа № 1.42. Адаптивный веб-дизайн Практическая работа № 1.43. Разработка протокола взаимодействия веб-сервисов Практическая работа № 1.44. Разработка REST API	ПК 1- ПК 5 ОК1-ОК10	Практическа я работа (отчет по практическо й работе)

	Практическая работа № 1.45. Разработка REST API		
Тема 1.10. Основы ADO.Net	<p>Практическая работа № 1.46. Теоретические основы Технологии ado.Net</p> <p>Практическая работа № 1.47. Создание базы данных в среде MssqlServerManagement</p> <p>Практическая работа № 1.48. Создание базы данных в среде MssqlServerManagement</p> <p>Практическая работа № 1.49. Копирование и восстановление базы данных</p> <p>Практическая работа № 1.50. Разработка формы для работы с данными в среде VsualStudio без кода</p> <p>Практическая работа № 1.51. СозданиеSql- запросов</p> <p>Практическая работа № 1.52. СозданиеSql- запросов в среде sql Server Management Studio</p> <p>Практическая работа № 1.53. Программирование с помощью встроенного языка transact sql в Microsoft Sql Server</p> <p>Практическая работа № 1.54. Разработка формы работы с магазинами с использованием объекта Command</p> <p>Практическая работа № 1.55. Разработка формы работы с магазинами с использованием объекта Command</p> <p>Практическая работа № 1.56. Создание, удаление и редактирование данных в отсоединенной среде</p> <p>Практическая работа № 1.57. Быстрое создание пользовательского интерфейса посредством связывания с данными</p> <p>Практическая работа № 1.58. Безопасность вMssqlServer</p> <p>Практическая работа № 1.59. Создание отчетных форм для баз данных средствами MsVisualStudio</p> <p>Практическая работа № 1.60. Самостоятельная работа по автономным и подключенным объектам</p> <p>Практическая работа № 1.61. Самостоятельная работа по автономным и подключенным объектам</p>	ПК 1- ПК 5 OK1-OK10	Практическая работа (отчет по практической работе)
МДК 01.02. Поддержка и тестирование программных модулей			
Тема 2.1 . Отладка и тестирование программного обеспечения	<p>Практическая работа № 2.1. Выявление ошибок и причин их появления</p> <p>Практическая работа № 2.2. Тестирование «белым ящиком»</p> <p>Практическая работа № 2.3. Тестирование «белым ящиком»</p>	ПК 1- ПК 5 OK1-OK10	Практическая работа (отчет по практической работе)

	<p>Практическая работа № 2.4. Тестирование «черным ящиком»</p> <p>Практическая работа № 2.5. Тестирование «черным ящиком»</p> <p>Практическая работа № 2.6. Модульное тестирование</p> <p>Практическая работа № 2.7. Интеграционное тестирование</p>		
<p>Тема 2.2. Основные принципы отладки и тестирования программных продуктов номер и наименование темы</p>	<p>Практическая работа № 2.8. Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования</p> <p>Практическая работа № 2.9. Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования</p> <p>Практическая работа № 2.10. Использование инструментальных средств на этапе отладки программного модуля</p> <p>Практическая работа № 2.11. Использование инструментальных средств на этапе отладки программного модуля</p>	<p>ПК 1- ПК 5 ОК1-ОК10</p>	<p>Практическая работа (отчет по практической работе)</p>
<p>Тема 2.3 Виды тестирования программных продуктов</p>	<p>Практическая работа № 2.12. Тестирование на этапе сопровождения программного продукта</p> <p>Практическая работа № 2.13. Введение Место верификации среди процессов разработки программного обеспечения</p> <p>Практическая работа № 2.14. Тестовые примеры. Классы эквивалентности. Ручное тестирование в MVSTE</p> <p>Практическая работа № 2.15. Тестовое окружение</p> <p>Практическая работа № 2.16. Модульное тестирование. Тестирование классов</p> <p>Практическая работа № 2.17. Автоматизация модульного тестирования</p> <p>Практическая работа № 2.18. Формальные инспекции</p> <p>Практическая работа № 2.19. Покрытие программного кода</p> <p>Практическая работа № 2.20. Повторяемость тестирования, зависимости тестовых примеров</p> <p>Практическая работа № 2.21. Интеграционное тестирование в MV STE</p> <p>Практическая работа № 2.22. Тестирование в Microsoft Solutions Framework</p>	<p>ПК 1- ПК 5 ОК1-ОК10</p>	<p>Практическая работа (отчет по практической работе)</p>
<p>Тема 2.4. Документирование</p>	<p>Практическая работа № 2.23. Оформление документации на программные средства</p>	<p>ПК 1- ПК 5 ОК1-ОК10</p>	<p>Практическая работа (отчет по</p>

	использованием инструментальных средств Практическая работа № 2.24. Оформление документации на программные средства с использованием инструментальных средств		практической работе)
МДК.01.03 Разработка мобильных приложений			
Тема 3.1. Мобильные приложения и технологии	Практическая работа № 3.1. Платформа Android Практическая работа № 3.2. Java 2 MicroEdition Практическая работа № 3.3. Протокол Bluetooth Практическая работа № 3.4. Установка инструментария для разработки мобильных приложений Практическая работа № 3.5. Настройка среды для разработки мобильных приложений Практическая работа № 3.6. Установка среды разработки мобильных приложений с применением виртуальной машины	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 3.2 Создание и тестирование модулей для мобильных приложений	Практическая работа № 3.7. Создание эмуляторов. Подключение устройств. Настройка режима терминала Практическая работа № 3.8. Создание нового проекта. Изучение кода. Комментирование кода. Изменение элементов дизайна Практическая работа № 3.9. Обработка событий: подсказки. Обработка событий: цветовая индикация. Подготовка стандартных модулей. Обработка событий: переключение между экранами Практическая работа № 3.10. Передача данных между модулями. Тестирование мобильного приложения. Оптимизация мобильного приложения	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 3.3. Разработка мобильных приложений под Android в среде eclipse и другими способами.	Практическая работа № 3.11. Создание тем для упрощения работы с элементами. Применение DDMS для отладки приложения. Создание логов. Списки. Работа с ориентацией экрана, применение различных layouts. Анимация. Рисование. Меню. Кнопки. Диалоговые окна. Сообщения. Мультимедиа. Shared preferences. БД SQLite. Интернет-соединение. Content providers. Работа с картами и GPS. Виджеты. Публикация приложения.	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 3.4. Разработка мобильных приложений под iOS.	Практическая работа № 3.12. Создание приложения, которое состоит из нескольких activities. Написание	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по

	приложение, работающее с разными темами/стилями. Создание приложения, содержащее анимированные интерфейсные элементы		практической работе)
Тема 3.5. Разработка мобильных приложений под Windows Phone.	<p>Практическая работа № 3.13. Создание приложения, отображающее после запуска карты Google или какие-нибудь другие карты. Создание собственного виджета с настройками. Создание приложения, использующего опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента</p> <p>Практическая работа № 3.14. Создание первого приложения под Android. Основы разработки интерфейсов мобильных приложений</p> <p>Практическая работа № 3.15. Создание многоэкранного приложения</p> <p>Практическая работа № 3.16. Демонстрации распознавания стандартных жестов. Принципы работы с жестами вводимыми пользователями</p> <p>Практическая работа № 3.17. Многооконное приложение. Геолокационные возможности. Использование сторонних библиотек</p> <p>Практическая работа № 3.18. Работа с базами данных в Android. Основные приемы работы с инструментами разработки</p> <p>Практическая работа № 3.19. Шаблоны проектов, структура проектов. Элементы управления. Разработка пользовательского элемента управления</p> <p>Практическая работа № 3.20. Навигация в приложении. Обмен данными внутри приложения</p> <p>Практическая работа № 3.21. Использование шаблона проектирования MVVM. Работа с JSON, XML, сжатие данных</p> <p>Практическая работа № 3.22. Работа WebClient и HttpRequest</p> <p>Практическая работа № 3.23. Работа с API веб-сервисов</p> <p>Практическая работа № 3.24. Хранение данных на устройстве. Локальные базы данных</p> <p>Практическая работа № 3.25. Разработка для Windows Azure</p> <p>Практическая работа № 3.26. Сервисы Live Connect: SkyDrive</p> <p>Практическая работа № 3.27. Многопоточное программирование</p>	ПК 1-ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)

	<p>Практическая работа № 3.28. Сенсорный пользовательский интерфейс</p> <p>Практическая работа № 3.29. Работа с датчиками, определение местоположения</p> <p>Практическая работа № 3.30. Распознавание и синтез речи, работа с камерой</p>		
МДК.01.04 Системное программирование			
Тема 4.1 Программирование на языке низкого уровня	<p>Практическая работа №4.1. Использование потоков.</p> <p>Практическая работа №4.2. Обмен данными.</p> <p>Практическая работа №4.3. Сетевое программирование сокетов.</p> <p>Практическая работа №4.4. Работы с буфером экрана.</p>	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 4.2. Программирование на Ассемблер	<p>Практическая работа №4.5. Команды пересылки данных общего назначения</p> <p>Практическая работа №4.6. Команды загрузки адресных значений и обращение к стеку</p> <p>Практическая работа №4.7. Команды ввода-вывода</p> <p>Практическая работа №4.8. Арифметические операции</p> <p>Практическая работа №4.9. Логические команды</p> <p>Практическая работа №4.10. Организация подпрограмм</p> <p>Практическая работа №4.11. Организация макрокоманд</p> <p>Практическая работа №4.12. Обработка двумерных массивов</p> <p>Практическая работа №4.13. Кросс-системы</p>	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)
Тема 4.3. Язык С#	<p>Практическая работа №4.14. Знакомство со средой разработки Visual С#.</p> <p>Структура программы на С#. Основы языка С#</p> <p>Практическая работа №4.15. Основы языка С#. Создание простой С#-программы</p> <p>Практическая работа №4.16. Основы языка С#. Создание и использование размерных типов данных</p> <p>Практическая работа №4.17. Основы языка С#. Использование выражений и исключений</p> <p>Практическая работа №4.18. Основы языка С#. Создание и использование методов</p> <p>Практическая работа №4.19. Основы языка С#. Создание и использование массивов</p>	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)

	<p>Практическая работа №4.20. Основы языка C#. Создание и использование классов</p> <p>Практическая работа №4.21. Основы языка C#. Создание и использование ссылочных переменных</p> <p>Практическая работа №4.22. Основы языка C#. Создание объектов и управление ресурсами</p> <p>Практическая работа №4.23. Основы языка C#. Использование наследования при реализации интерфейсов</p> <p>Практическая работа №4.24. Основы языка C#. Использование модификатора доступа internal и создание сборок</p>		
Тема 4.4. Язык Python	<p>Практическая работа №4.25. Переменные, арифметические операции, функции print и input, логический тип bool и операторы сравнения</p> <p>Практическая работа №4.26. Работа со строками, форматирование строк, списки и их срезы, вложенные списки</p> <p>Практическая работа №4.27. Условный оператор, тернарный условный оператор</p> <p>Практическая работа №4.28. Операторы циклов, вложенные циклы и итерируемые объекты</p> <p>Практическая работа №4.29. Генераторы списков</p> <p>Практическая работа №4.30. Словари, кортежи, множества</p> <p>Практическая работа №4.31. Разработка игрового приложения</p> <p>Практическая работа №4.32. Разработка игрового приложения</p> <p>Практическая работа №4.33. Разработка игрового приложения</p>	ПК 1- ПК 5 ОК1-ОК10	Практическая работа (отчет по практической работе)

2. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ

Текущий контроль осуществляется по результатам выполнения практических работ, представленных в методических указаниях, которые содержат:

- Практические работы
- Критерии оценивания
- Вопросы для самопроверки
- Перечень рекомендуемой литературы

МДК 01.01. Разработка программных модулей

Примеры тестовых заданий для текущего контроля

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. Бесбойность 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. рентабельность
20. Возможность «параллельного» использования несколькими процессами соответствует характеристике эксплуатационных требований ...
- a. универсальность
 - b. программная совместимость
 - c. вповторная входимость
 - d. рентабельность
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. Идентификатором, сформированным по всем правилам языка C, можно назвать выражение ...
 - a. lizonbka
 - b. Uzo4ka_
 - c. nonaroshky
 - d. egoza-liza
- 34. Для описания основных (стандартных) типов данных на языке C определены следующие ключевые слова...
 - a. int, char, bool, double, void
 - b. int, char, float, double, bool
 - c. int, wchar_t, float, double, char
 - d. int, char, float, double, void
- 35. 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
36. В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...
- a. Формы
 - b. Инспектора объектов
 - c. Кода
 - d. Деревя объектов
37. В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...
- a. Caption
 - b. Text
 - c. Align
 - d. Name
38. 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
39. 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
40. 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
41. 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
42. 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 до-статочно мало ошибок
43. 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
44. 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;
45. 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;
46. 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;
47. Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...
- a. private
 - b. protected
 - c. public
 - d. published
48. Видимость переменных и методов класса в классах потомков определяет спецификатора доступа ...
- a. private
 - b. protected
 - c. public
 - d. published
49. Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...

- a. private
 - b. protected
 - c. public
 - d. published
50. Процесс, в ходе которого один объект может приобретать свойства другого, называется...
- a. наследование
 - b. инкапсуляция
 - c. полиморфизм
 - d. динамический полиморфизм
51. Процесс перегрузки функций и операторов реализует принцип ...
- a. наследования
 - b. инкапсуляции
 - c. статического полиморфизма
 - d. динамического полиморфизма
52. На основе наследования и виртуальных функций реализуется принцип ...
- a. наследования
 - b. инкапсуляции
 - c. статического полиморфизма
 - d. динамического полиморфизма

Теоретические задания

1. Унифицированный язык моделирования (UML). Определение прецедентов (вариантов использования). Построение концептуальной модели предметной области. Описание поведения системы с помощью диаграмм последовательности действий и диаграмм взаимодействий.
2. Проектирование программного обеспечения при структурном подходе. Структурная схема разрабатываемого программного обеспечения (логическая и физическая модели) Метод пошаговой детализации при составлении алгоритмов.
3. Основы файловой системы. Порядок работы с текстовыми и двоичными файлами. Функции для работы с файлами разного типа. Правила бесформатного и форматного ввода/вывода.
4. Проектирование программного обеспечения при объектном подходе. Структура проекта в объектно-ориентированной среде программирования. Событийно-управляемая модель программирования. Компоненты объектноориентированной среды. Принципы разработки графического интерфейса. Построение функциональной блок-схемы приложения. Особенности ввода/вывода в объектно-ориентированной среде программирования. Виды строк, функции преобразования типов.
5. Виды программных документов. Пояснительная записка. Руководство пользователя. Руководство системного программиста.
6. Тестирование методами «белого ящика» и «черного ящика». Порядок разработки тестов. Автоматизация тестирования. Модульное, интеграционное, системное тестирование.
7. Вставка, добавление и удаление записей таблиц базы данных. Поиск, фильтрация и сортировка данных. Возможности среды для создания новых полей.

МДК 01.02. Поддержка и тестирование программных модулей

Примеры тестовых заданий для текущего контроля

Тест 1: Модульное и интеграционное тестирование

<https://videouroki.net/tests/834491111/>

Список вопросов теста

- Вопрос 1. Какие существуют особенности управляющего графа программы в случае объектно-ориентированного программирования (ООП)?
- она становится неприменимой
 - она требует адаптации по обработке сообщений
 - она требует описания поведения программы
 - она требует описания не только структуры, но и поведения программы
- Вопрос 2. Какие существуют особенности интеграционного тестирования для ООП?
- тестирование дерева классов
 - тестирование последовательностей прямых вызовов методов с помощью R-путей
 - тестирование последовательностей обработки сообщений с помощью MM-путей
 - тестирование исключительно цепочек вызовов процедур программ
- Вопрос 3. Какие этапы включает методика ООП при тестировании программного комплекса?
- тестирование методов каждого класса программного комплекса
 - тестирование отношений между классами с помощью тестов на основе R-путей или MM-путей
 - тестирование взаимодействия модулей по всей иерархии комплекса
- Вопрос 4. Какие этапы методики тестирования используются в ООП?
- тестирование методов каждого класса программы
 - тестирование методов класса, входящих в его контекст
 - тестирование дерева классов программного проекта, включающего оттестированный класс
- Вопрос 5. Какие этапы методов тестирования класса используются в ООП?
- тестирование класса как модуля по выбранному критерию
 - тестирование класса как иерархической структуры
 - тестирование классов, входящих в модель проекта
- Вопрос 6. Что такое управляющий граф программы (УГП)?
- множество операторов программы
 - граф, вершины которого кодируют операторы программы, а дуги - управления (порядок исполнения) операторов
 - множество операторов управления
- Вопрос 7. Что такое путь в УГП?
- последовательность вершин и дуг управляющего графа программы с фиксированными начальной и конечной вершиной
 - последовательность ветвей управляющего графа программы с фиксированными начальной вершиной первой ветви и конечной вершиной последней ветви пути
 - множество связанных дуг управляющего графа программы
- Вопрос 8. Что такое ветвь управляющего графа программы?
- 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
 - часть пути, в котором все внутренние вершины кодируют линейные операторы
 - начальная и конечная вершина пути
- Вопрос 9. Можно ли гарантировать остановку программы на любом тесте?
- в общем случае нет
 - возможно в частных случаях
 - задача в общей постановке алгоритмически неразрешима
- Вопрос 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

Вопрос 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

Вопрос 12. Каковы фазы процесса построения тестовых путей?

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

Какие существуют методы построения тестовых путей?

- a. статические
- b. динамические
- c. методы реализуемых путей

Вопрос 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

Вопрос 15. Какие существуют разновидности интеграционного тестирования?

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

Вопрос 16. Каковы особенности нисходящего тестирования?

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

Вопрос 17. Каковы особенности восходящего тестирования?

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

Вопрос 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

Вопрос 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

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

- a. никогда
- b. да, всегда

- c. возможно в отдельных случаях

Тест 2: Системное и регрессионное тестирование

<https://videouroki.net/tests/537207033/>

Список вопросов теста

Вопрос 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»

Вопрос 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

Вопрос 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

Вопрос 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

Вопрос 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

Вопрос 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

Вопрос 7. Можно ли гарантировать безопасность метода регрессионного тестирования при отсутствии информации об изменениях в программе а. нет b. да

Вопрос 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

Вопрос 9. Какими свойствами обладает метод повторного прогона всех тестов? а. полнота b. точность c. эффективность d. универсальность

Вопрос 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

Вопрос 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

Вопрос 12. Какие из перечисленных методов тестирования дают наиболее надежные результаты? а. статические методы b. модульное тестирование c. интеграционное тестирование d. системное тестирование с моделируемым окружением e. системное тестирование в реальном окружении и реальном времени

Вопрос 13. Что влияет на обеспечение качества программного продукта? а. тестирование b. анализ дизайна c. обзоры кода d. аудиты процесса разработки e. ревьюирование проектной документации

Вопрос 14. Какие основные уровни выделены в процессе тестирования?

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

Вопрос 15. 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

Вопрос 16. 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

Вопрос 17. 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

Вопрос 18. 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

Время выполнения: 20 мин

Количество вариантов - 20

Критерии оценки:

Оценка	Процент выполнения заданий теста
5	85
4	70
3	50
2	Менее 50

МДК.01.03 Разработка мобильных приложений

Примеры тестовых заданий для текущего контроля

1. Какие методы отражают жизненный цикл Activity:
 - a. onCreate() (+)
 - b. onRestart()(+)
 - c. onResume()(+)
 - d. onRunning()
 - e. onClose()
2. Подробная информация о приложении содержится в файле:
 - a. AndroidManifest.xml (+)
 - b. main.xml
 - c. R.java
 - d. default.properties
3. Для создания всплывающего уведомления необходимо инициализировать объект:
 - a. Toast (+)
 - b. Message
 - c. TextView
 - d. MessageBox
4. Единица измерения dp или dip - это 1/72 дюйма,
 - a. определяется по физическому размеру экрана дюйм,
 - b. определяется по физическому размеру экрана абстрактная ЕИ,
 - c. позволяющая приложениям выглядеть одинаково на различных экранах и разрешениях (+)
 - d. физический элемент матрицы дисплея
5. Тип верстки при котором позиционирование элементов происходит относительно друг друга и относительно главного контейнера
 - a. AbsoluteLayout
 - b. FrameLayout
 - c. LinearLayout
 - d. RelativeLayout (+)
6. Какой класс можно использовать для перехода между Activity?

- a. ActivityChanger
 - b. Activity Intent (+)
 - c. Switcher
7. От какого класса наследуются все элементы управления?
- a. Control
 - b. Controls
 - c. Element
 - d. View (+)
8. Для чего используется класс R?
- a. Класс ресурсов (+)
 - b. Класс для управления элементами управления
 - c. Класс для работы с потоками
 - d. Класс, предназначенный только для доступа к идентификаторам элементов
9. Как программно получить доступ к элементам управления через идентификатор?
- a. getViewById(id)
 - b. findViewById(id) (+)
 - c. findViewById(id)
 - d. getView(id)
10. В какой библиотеке находится класс Activity?
- a. android
 - b. android.Activities
 - c. android.app (+)
 - d. java.app

- 11 На основе какой операционной системы была разработана система Android:
- a. Linux (+)
 - b. MiniX
 - c. Windows
 - d. iOS
- 12 Как называется режим использования Android-системы с максимальными правами (аналог администратора Windows):
- a. Суперпользователь (+)
 - b. Root-Home пользователь
 - c. Android user
 - d. Up-User
- 13 Как называется событие-клик объекта-кнопки:
- a. OnClick (+)
 - b. OnClicked
 - c. Click
 - d. ClickEvent
- 14 Для добавления в приложение более одного экрана взаимодействия необходимо создание:
- a. Activity (+)
 - b. View
 - c. Layout
 - d. XML-файла
- 15 Как называется класс для работы с SQLite-бд¹ами в Android: SQLiteOpenHelper
- a. (+)
 - b. SQLProvider
 - c. SQLiteProvider
 - d. SQLiteContext
- 16 Как называется язык разметки Window Phone приложений: XAML (+)
- a.
 - b. XML
 - c. Java
 - d. C#
- 17 С помощью какого класса можно вывести сообщение в Window Phone приложении:
- a. Toast
 - b. MessageBox (+)
 - c. Message
 - d. ShowClass
- 18 Какой элемент предоставляет вариант разметки, элементы, помещенные в который, располагаются в стек горизонтально или вертикально:
- a. StackPanel (+)
 - b. Pivot
 - c. Panorama
 - d. Canvas
- 19 Какой элемент позволят позиционировать элементы внутри себя максимально гибко предоставляя возможность размещать элементы, используя строки и столбцы.
- a. Grid (+)
 - b. Pivot
 - c. Panorama
 - d. Canvas

20. Какой элемент предоставляет наиболее простой вариант разметки. Он может быть использован для абсолютного позиционирования элементов с использованием координат.
 - a. Grid
 - b. Pivot
 - c. Panorama
 - d. Canvas (+)
21. На базе какого ядра сделано ядро Android?
 - a. Windows NT
 - b. Linux
 - c. FreeBSD
 - d. VxWorks
22. Какое название носит виртуальная Java-машина Android?
 - a. Dalvik
 - b. B ionic
 - c. HotSpot
23. Как изменился набор консольных утилит Linux в Android?
 - a. Существенно уменьшился
 - b. Остался без изменений
 - c. Существенно увеличился
 - d. В Android нет консольных утилит
24. Для чего предназначен Binder?
 - a. Связывает файлы ресурсов с приложением
 - b. Обеспечивает межпроцессное взаимодействие
 - c. Обмен информацией между приложениями
25. Является ли logcat полным аналогом gdb?
 - a. Да
 - b. logcat — полный аналог gdbserver
 - c. Нет
26. Из-за чего glibc была заменена на Bionic?
 - a. Смена бренда
 - b. Несовместимость glibc со свободными лицензиями
 - c. glibc невозможно скомпилировать для ARM
 - d. Уменьшение размеров объектного кода
27. Какой байт-код используется в Dalvik?
 - a. Обычный
 - b. Исключительно свой собственный
 - c. Свой собственный, но обычный может быть в него сконвертирован
28. Из чего состоит Java API в Android?
 - a. Часть стандартных библиотек Java
 - b. Стандартные библиотеки Java и пакет javaх.
 - c. Набор стандартных, популярных и собственных пакетов
 - d. Исключительно собственные библиотеки Android
29. Какую функциональность обеспечивает пакет android.view?
 - a. Реализация GUI
 - b. Работа с камерой
 - c. Темы рабочего стола
30. Какой способ беспроводной связи есть в пакете android.net?
 - a. BlueTooth
 - b. WiFi
 - c. NFC
 - d. DRM

Примерные теоретические вопросы

1. Архитектура платформы Android. Уровень ядра. Уровень библиотек.
2. Архитектура платформы Android. Dalvik Virtual Machine.
3. Архитектура платформы Android. Уровень каркаса приложений. Уровень приложений.
4. Среда разработки для Android. Eclipse IDE. Плагин ADT. Android Virtual Device.
5. Android SDK. Версии SDK и Android API Level.
6. Структура проекта Android-приложения в Eclipse. Каталоги ресурсов. Файл R.java.
7. Графический интерфейс пользователя в Android-приложениях. XMLpa3MeiKa интерфейса.
8. Архитектура платформы Android.
9. XMI .-разметка интерфейса пользователя.
10. XAMI .-разметка интерфейса пользователя.
11. Базовые элементы управления.
12. Ресурсы в Android-приложениях.
13. Ресурсы в Windows Phone-приложениях.
14. Активности и интенды.
15. Обработка пользовательского ввода. Касания, ввод текста.
16. Типы компоновок графического интерфейса. FraneLayout, LinearLayout, TableLayout, RelativeLayout.
17. Базовые элементы управления. TextView. EditText. Тип ввода текста. Параметры отображения клавиатуры. ImageView.
18. Диалоговые окна. AlertDialog. ProgressDialog. DatePickerDialog. TimePickerDialog. Создание пользовательских диалоговых окон.
19. Многопоточные приложения в Android и Windows Phone. Использование системных таймеров и системного времени.
20. Процессы в Android. Объекты Activity. Состояния Activity.
21. Использование объектов Intent. Intent-фильтры.
22. Использование ресурсов. Ссылки на ресурсы. Загрузка простых типов из ресурсов. Загрузка файлов произвольного типа.
23. Файловая система Android. Чтение и запись файлов.
24. Адаптеры данных. Отображение данных в компонентах ListView, GridView, AutoCompleteTextView, MultiAutoCompleteTextView.
25. Пользовательские настройки. Использование SharedPreferences. Виды настроек.
26. Работа с графикой. Drawable и Canvas.
27. Работа с анимацией. Tween Animation и Frame Animation. Описание анимации в XML и в коде программы.
28. Службы в Android. Компонент Service.
29. Датчики мобильных устройств. Управление датчиками в приложении. Виды датчиков и особенности их использования.
30. Программный доступ к дисплею устройства. Менеджер окон. Параметры дисплея.

Примеры практико-ориентированных заданий

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),

выделив все инструменты для работы с ней; сконфигурировать эмулятор (желательно эмулировать своё собственное мобильное устройство, включая версию андроид на нём в качестве Target SDK). Изменить TextView с надписью "Hello world" на какую-нибудь другую

надпись по желанию. Запустить проект на эмуляторе и убедиться, что всё работает.

Задание 2. Создать новый проект, написать программу, которая выводит в элемент TextView надпись, введённую пользователем в текстовом поле EditText после нажатия на кнопку Button. Помимо этого в Activity должен быть TextView с ФИО студента и группой. Запустить на эмуляторе и убедиться, что всё работает.

Задание 3. Создать приложение, которое состоит из нескольких activities. Первое activity содержит элемент TextView с названием или номером activity, текстовое поле EditText для ввода какой-то информации, кнопку Button с названием "Next" или "Перейти на 2 activity/экран/окно" или просто "2". Помимо этого в 1 activity должен быть TextView с ФИО студента и группой. После нажатия на эту кнопку происходит переход на второе activity, где содержится TextView с названием или номером activity, TextView с надписью что-то вроде "В первом окне вы напечатали:" и под ним - ещё один TextView с содержимым EditText с первого activity, и, разумеется, кнопка "1" или "Вернуться на 1 экран" или "Вернуться к вводу текста", нажав на которую пользователь может перейти обратно к 1 activity. Запустить на эмуляторе и убедиться, что всё работает.

Задание 4. В новом проекте написать приложение, работающее с разными темами/стилями. Сначала создать свой стиль и применить его к какому-нибудь интерфейсному элементу, затем - свою тему, которая применяется ко всем интерфейсным элементам. Приложение при этом должно выглядеть нестандартно, запустить на эмуляторе и убедиться, что всё работает. При возникновении ошибок открыть лог (CatLog) внизу, найти первую красную надпись и породить свою тему от той, которая требуется в этом красном сообщении.

Задание 5. Создать пользовательский (свой) список. Например, получить доступ в приложении к контактам (Permissions-закладка в AndroidManifest.xml) и скопировать контакты телефона в свой список, который отобразить после запуска приложения. Или создать свой список в виде твиттера (картинка+текст), элементы которого просто статически задать в массиве (как и картинки).

Задание 6. Создать приложение, содержащее анимированные интерфейсные элементы (например, увеличивающиеся при клике на них кнопки, вращающиеся TextView и т.д.).

Задание 7. Создать приложение, отображающее после запуска карты Google или какие-нибудь другие карты.

Задание 8. Создать собственный виджет с настройками. Например, виджет, который открывает какой-то сайт, адрес которого можно поменять в настройках.

Задание 9. 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 10. 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 11. Создать приложение, помещающее по нажатию на кнопку какое-то сообщение со звуком в панель уведомлений/статус-панель на эмуляторе.

Задание 12. Создать приложение, по нажатию кнопки в котором проигрывается какой-то звук.

Задание 13. Создать приложение, при запуске которого проигрывается какое-то видео.

Задание 14. Создать приложение, при запуске которого активируется фотокамера телефона, производится снимок, и этот снимок помещается в ImageView интерфейса приложения.

Задание 15. Создать приложение, работающее с SharedPreferences и сохраняющее настройки, а также работающее с БД SQLite - заполняющее БД по нажатию кнопки 1 с помощью EditText, и выводящее вес записи этой БД в какой-нибудь интерфейсный элемент

ниже с помощью кнопки 2 (в виде списка, datagrid или просто правильно настроенного Text View).

МДК.01.04 Системное программирование

Тестовые задания для текущего контроля

Раздел 1. Введение в системное программирование

1. Операционная система:
 - a. система программ, которая обеспечивает совместную работу всех устройств компьютера по обработке информации
 - b. система математических операций для решения отдельных задач
 - c. система планового ремонта и технического обслуживания компьютерной техники
2. Программное обеспечение (ПО) - это:
 - a. совокупность программ, позволяющих организовать решение задач на компьютере
 - b. возможность обновления программ за счет бюджетных средств
 - c. список имеющихся в кабинете программ, заверен администрацией школы
3. Загрузка операционной системы - это:
 - a. запуск специальной программы, содержащей математические операции над числами
 - b. загрузка комплекса программ, которые управляют работой компьютера и организуют диалог пользователя с компьютером
 - c. вложение дискеты в дисковод
4. Система программирования - это:
 - a. комплекс любимых программ программиста
 - b. комплекс программ, облегчающий работу программиста
 - c. комплекс программ, обучающих начальным шагам программиста
5. Прикладное программное обеспечение - это:
 - a. справочное приложение к программам
 - b. текстовый и графический редакторы, обучающие и тестирующие программы, игры
 - c. набор игровых программ
6. Прикладное программное обеспечение:
 - a. программы для обеспечения работы других программ
 - b. программы для решения конкретных задач обработки информации
 - c. программы, обеспечивающие качество работы печатающих устройств
7. Операционные системы:
 - a. DOS, Windows, Unix
 - b. Word, Excel, Power Point
 - c. (состав отделения больницы): зав. отделением, 2 хирурга, 4 мед. Сестры
8. Системное программное обеспечение:
 - a. программы для организации совместной работы устройств компьютера как единой системы
 - b. программы для организации удобной системы размещения программ на диске
 - c. набор программ для работы устройства системного блока компьютера
9. Сервисные (обслуживающие) программы:
 - a. программы сервисных организаций по бухгалтерскому учету
 - b. программы обслуживающих организаций по ведению делопроизводства

- c. системные оболочки, утилиты, драйвера устройств, антивирусные и сетевые программы
10. Системные оболочки - это:
- a. специальная кассета для удобного размещения дискет с операционной системой
 - b. специальная программа, упрощающая диалог пользователь - компьютер, выполняет команды операционной системы
 - c. система приемов и способов работы конкретной программы при загрузке программ и завершении работы
11. Использование одного имени для задания общих для класса действий, что означает способность объектов выбирать внутренний метод, исходя из типа данных, определяет свойство ООП
- a. Полиморфизм
 - b. Управление событиями
 - c. Инкапсуляция
 - d. Наследование
12. Понятие «инкапсуляция» относится к
- a. Технологии модульного программирования
 - b. Технологии объектно - ориентированного программирования
 - c. Технологии императивного программирования
 - d. Технологии модульного программирования
13. Свойство ООП, которое может быть смоделировано с помощью таксономической классификационной схемы (иерархии) называется
- a. Инкапсуляция
 - b. Управление событиями
 - c. Полиморфизм
 - d. Наследование
14. Понятие класса в ООП включает в себя
- a. Поля и методы класса
 - b. Процедуры и функции обработки
 - c. Поля и функции обработки
 - d. Поля и процедуры обработки
15. Назначение конструктора объекта
- a. Только выделяет память под объект
 - b. Выделяет память и задает начальное значение полям
 - c. Задает начальное значение полям
 - d. Выделяет память, задает начальное значение полям, выполняет любые проверки, заданные программистом
16. Как описывается конструктор объекта
- a. procedure create;
 - b. constructor create;
 - c. function create;
 - d. function constructor;
17. Как описывается деструктор объекта
- a. procedure free;
 - b. destructor free;
 - c. free;
 - d. function free;
18. Понятие объекта в ООП - это
- a. представитель класса
 - b. конкретные данные, заданные в классе.
 - c. компонент панели инструментов
 - d. встроенный объект Delphi

- 19 . Моделями типа «черный ящик» являются
- Модели мышления
 - Модели, описывающие зависимость параметров состояния объекта от входных параметров
 - Модели, описывающие входные и выходные параметры объекта без учета внутренней структуры объекта
 - Модели «аварийного» ящика на самолетах
- 20 . Моделями типа «белый ящик» являются
- Модели мышления
 - Модели, описывающие зависимость параметров состояния объекта от входных параметров
 - Модели, описывающие входные и выходные параметры объекта с учетом внутренней структуры объекта
 - Модели, описывающие выходные данные в программе
- 21 . Программа «драйвер» служит для
- запуска программы на выполнение
 - имитации запуска программы на выполнение
 - проверки правильности работы программы
 - передачи параметров в процедуры и функции
- 22 . Программа «заглушка» служит для
- запуска программы на выполнение
 - имитации запуска другой программы на выполнение
 - проверки правильности работы программы
 - имитации передачи параметров в другой модуль
- 23 .Какие методы сборки программы существуют а. монолитная б. пошаговая с. одновременная d. постепенная
- 24 .Какой метод тестирования программы учитывает закон распределения входных данных
- детерминированное тестирование
 - функциональное тестирование
 - стохастическое тестирование
 - логическое тестирование
- 25 .Программирование сверху вниз - это
- Процесс, при котором от начального предположения осуществляется движение по направлению к лучшим решениям
 - Процесс пошагового разбиения алгоритма на все более мелкие части с целью получения таких элементов, для которых можно написать конкретные команды
 - Метод сведения трудной задачи к последовательности более простых
 - Исследование древовидной модели пространства решений и ориентация на поиск оптимального решения
26. Загрузочный модуль программы - результат работы
- Грамматики
 - Транслятора
 - Интерпретатора
 - Редактора связей (компоновщика)
- 27 .Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется
- построителем кода
 - компилятор
 - переводчиком
 - преобразователем
- 28 .Результатом компиляции программы на языке высокого уровня является

- a. Командный файл
- b. Объектный файл
- c. Исходный текст программы на языке высокого уровня
- d. Дисплейный файл

Раздел 2. Основы Ассемблер

1. Установите порядок, в котором осуществляется перевод произвольного двоичного числа в системе счисления с основанием $q=2^n$. 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 $q = 2^n$; 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 празрядов, 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 100011111011 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 $q=2^n$. Укажите порядок следования всех 3 вариантов ответа: 1. данное двоичное число разбить справа налево на группы по пцифр в каждой; 2. рассмотреть каждую группу как n -разрядное двоичное число и записать ее соответствующей цифрой в системе счисления с основанием $q = 2^n$; 3. если в последней левой группе окажется меньше раз рядов, 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
- 1111001011011001
 - 1111101011001100
 - 1011010111111100
 - 1101100111011010
9. Выполните вычисление в двоичной системе счисления $110101-101=$
- 110100
 - 110010
 - 110000
 - 110001
10. В саду 100 фруктовых деревьев - 14 яблонь и 42 груши. В какой системе счисления посчитаны деревья?
- В шестеричной
 - В шестнадцатеричной
 - В двоичной
 - В восьмеричной
11. Умножьте в двоичной системе счисления $1111*11=$
- 1111111
 - 101111
 - 1010101
 - 101101
12. Установите соответствие. Укажите соответствие для всех 4 вариантов ответа:
- 1) 4 2) 5 3) 6 4) 7
- 4=101 5=110 6=100 7=111
 - 5=101 6=110 4=100 7=111
 - 6=101 5=110 4=100 7=111
 - 4=101 7=110 6=100 5=111
13. Выполните умножение в системе счисления $10101*101=$
- 110101011
 - 1101001
 - 111001001
 - 1010101
14. Установите соответствие. Укажите соответствие для всех 4 вариантов ответа:
- A=1010 B=1111 F=1110 E=1011
 - A=1010 F=1111 E=1110 B=1011
 - A=1010 C=1111 F=1110 E=1011
 - A=1010 B=1111 F=1110 C=1011
15. В какой системе счисления будет верным равенство $7+8=16?$
- 12
 - 6
 - 9
 - 16
16. $FFFF+1=$
- 1FFF
 - 10000
 - FFFE
 - FFF1
17. Выполните вычитание в двоичной системе счисления $1111-101$
- 1010
 - 1110
 - 1100
 - 1001

18. Результат деления в двоичной системе счисления числа 110 на 11 равен
- 11
 - 10
 - 01
 - 101
19. Перевести смешанное двоичное 111100000,101 число в восьмеричную систему счисления
- 610,4
 - 740,5
 - 170,5
 - 1E0,A
20. Переведите восьмеричное число 0,25 в двоичную систему счисления
- 0,010101
 - 0,101010
 - 0,110111
 - 0,101011
21. Алгоритм-это:
- Указание на выполнение действий,
 - Система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи,
 - Процесс выполнения вычислений, приводящих к решению задачи
22. Свойство алгоритма - дискретность, выражает, что:
- Команды должны следовать последовательно друг за другом,
 - Каждая команда должна быть описана в расчете на конкретного исполнителя,
 - Разбиение алгоритма на конечное число команд
23. Формальное исполнение алгоритма - это:
- Исполнение алгоритма конкретным исполнителем с полной записью его рассуждений,
 - Разбиение алгоритма на конкретное число команд и пошаговое их исполнение,
 - Исполнение алгоритма не требует рассуждений, а осуществляется исполнителем автоматически
 - Исполнение алгоритма осуществляется исполнителем на уровне его знаний
24. Самое важное свойство алгоритма:
- Визуальность,
 - Массовость,
 - Дискретность,
 - Аудиальность,
25. Какой алгоритм называется линейным:
- Выполнение операций зависит от условия,
 - Операции выполняются друг за другом,
 - Одни и те же операции выполняются многократно
 - Присутствие всех возможных операций в одном алгоритме
26. Графическое задание алгоритма - это:
- Способ представления алгоритма с помощью геометрических фигур,
 - Представление алгоритма в форме таблиц и расчетных формул,
 - Система обозначений и правил для единообразной и точной записи алгоритмов и их исполнения
27. В расчете на кого должен строиться алгоритм:
- В расчете на ЭВМ,
 - В расчете на умственные способности товарища,
 - В расчете на конкретного исполнителя
28. Какое из перечисленных свойств относится к свойствам алгоритма:
- 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);
31. Псевдокоманда DD 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);
32. К сегментным регистрам относят
- a. CS, DS, SS и ES.
 - b. AX, BX, CX и DX
 - c. SP и BP.
 - d. SI и DI
33. К индексным регистрам относят
- a. CS, DS, SS и ES.
 - b. AX, BX, CX и DX
 - c. SP и BP.
 - d. SI и DI
34. Устанавливает бит результата в 1, если оба бита, бит источника и бит приемника установлены в 1.
- a. AND
 - b. OR
 - c. XOR (НЕ ИЛИ)
 - d. NOT
35. устанавливает бит результата в 1, если бит источника отличается от бита приемника.
- a. AND
 - b. OR
 - c. XOR
 - d. NOT
36. Запись регистра в память осуществляется при помощи команды
- a. MOV
 - b. XCHG
 - c. XLAT
37. Порядковый номер элемента в массиве, который всегда начинается с нуля
- a. Индекс
 - b. Команда
 - c. Псевдокоманда
 - d. Макрокоманда
38. Позволяет сохранять длину исходного текста программы

- a. Индекс
- b. Команда
- c. Псевдокоманда
- d. Макрокоманда

Примерные теоретические вопросы

1. Объясните, как Вы понимаете термин системное программное обеспечение сформулируете основные понятия.
2. Расскажите, как реализуются арифметические операции на Ассемблере.
3. Объясните, как Вы понимаете команды обработки строковых данных.
4. Перечислите и охарактеризуйте средства аппаратной поддержки функций ОС.
5. Объясните двоичное кодирование информации. Представление элементарных типов данных: натуральные числа, целые числа со знаком, числа с плавающей точкой.
6. Расскажите про компилятор ассемблер - программ, редактор связей (загрузчик).
7. Объясните общую структуру машинных команд.
8. Как организуется ввод - вывод и классифицируются внешние устройства.
9. Назовите и опишите константы, метки и условную компиляцию.
10. Перечислите и опишите этапы проектирования и выполнения программ
11. Опишите задачи, для решения которых используется язык ассемблера.
12. Объясните регистры еах, еbх, есх, еdх и их специальные свойства.
13. Расскажите про команды ADC и SBB.
14. Дайте определения резидентным программам.
15. Расскажите про команды MUL и IDIV.
16. Опишите уровни сложности резидентных программ
17. Расскажите про команды IMUL и DIV.
18. Опишите операнды Ассемблера
19. Опишите назначение простейших команд Ассемблера
20. Дайте характеристику языку ассемблера, определение, преимущества.

Примеры практико-ориентированных заданий

1. Описать переменные типа DB с инициализацией 123Gh.
2. Описать переменные типа DW с инициализацией ABCD4h
3. Описать переменные типа DD с инициализацией 6789h
4. Описать переменные типа DB, DW, DD без инициализации.
5. Описать одномерный массив DB 12 элементов с инициализацией на 0.
6. Описать одномерный массив DW 30 с инициализацией на ?.
7. Описать одномерный массив DD из 10 элементов с инициализацией на 12Ah.
8. Описать одномерный массив DB, DW, DD без инициализации.
9. Записать данные по физическим адресам ABCDEh, 525h.
10. Записать данные по физическим адресам AB34Eh, 52R5h.
11. Записать данные по физическим адресам 567DEh, AS525h.
12. Записать данные по физическим адресам 345Dh, A525Dh.
13. Записать данные по физическим адресам A1h, DF5h.
14. Записать число A034h в памяти с использованием прямой и косвенной адресации.
15. Записать число D2334h в памяти с использованием прямой и косвенной адресации.
16. Записать число C044h в памяти с использованием прямой и косвенной адресации.
17. Записать число AC34h в памяти с использованием прямой и косвенной адресации.
18. Сохранить регистр BX в памяти с использованием прямой и косвенной адресации.
19. Сохранить регистр AX в памяти с использованием прямой и косвенной адресации.

20. Сохранить регистр RX в памяти с использованием прямой и косвенной адресации.
21. Сохранить регистр FX в памяти с использованием прямой и косвенной адресации.
22. Сохранить регистр SX в памяти с использованием прямой и косвенной адресации.
23. Сохранить регистр HX в памяти с использованием прямой и косвенной адресации.
24. Сохранить регистр CX в памяти с использованием прямой и косвенной адресации.
25. Скопировать слово в память с использованием прямой и косвенной адресации.
26. Скопировать двойное слово в память с использованием прямой и косвенной адресации.
27. Скопировать байт в память с использованием прямой и косвенной адресации.
28. В памяти имеется массив байт со значениями 1, 5, 16, 26, 33. Вычислить таблично $y=x+3$ для всех элементов этого массива.
29. В памяти имеется массив байт со значениями 1, 6, 23, 24, 43. Вычислить таблично $b=a+4$ для всех элементов этого массива.
30. В памяти имеется массив байт со значениями 1, 7, 18, 24, 31. Вычислить таблично $c=v+5$ для всех элементов этого массива.

Раздел 3. Программирование на языке высокого уровня C

№ п/п	Задание Блока А (вопрос - тест)	Эталон ответа
Инструкция по выполнению заданий № 1-18: Выберите букву, соответствующую правильному варианту ответа и запишите её в бланке ответов.		
1.	Типы данных в языке C# принято классифицировать как: (Ответ считается верным, если отмечены все правильные варианты ответов) А) фигурные В) определенные программистом	B,C,D,E

	<p>С) простые D) встроенные (базисные) E) сложные (структурные) F) правильных ответов нет</p>	
2.	<p>Укажите правильный порядок следования приоритетов бинарных операций: (Отметьте один правильный вариант ответа.) A) арифметические, логические, отношения B) отношения, логические, арифметические C) арифметические, отношения, логические D) правильных ответов нет</p>	С
3.	<p>Отличительной особенностью каких языков программирования является их ориентация не на систему команд той или иной ЭВМ, а на систему операторов, характерных для записи определенного класса алгоритмов? (Отметьте один правильный вариант ответа.) A) языков программирования низкого уровня B) языков программирования высокого уровня C) языков программирования сверхвысокого уровня D) правильных ответов нет</p>	В
4.	<p>Что используют все языки программирования высокого уровня для предоставления программисту простого и легкого доступа к различным объектам? (Отметьте один правильный вариант ответа.) A) идентификаторы B) коммутаторы C) спецификаторы D) правильных ответов нет</p>	А
5.	<p>Что понимают под языком программирования (ЯП)? A) язык, предназначенный для решения определенного класса задач (проблем) B) правила представления данных и записи алгоритмов их обработки, которые автоматически выполняются ЭВМ C) язык, предназначенный для создания пакетов прикладных программ, в том числе для современных операционных систем D) правильных ответов нет</p>	В
6.	<p>В результате выполнения фрагмента программы <code>double x = 0, y = 0, z = x/y;</code> (Отметьте один правильный вариант ответа.) A) ошибки не будет и значение переменной z будет равно null B) возникнет ошибка на этапе компиляции программы C) возникнет ошибка на этапе выполнения программы D) ошибки не будет и значение переменной z будет равно Infinity E) ошибки не будет и значение переменной z будет равно NaN F) ошибки не будет и значение переменной z будет равно 0 G) правильных ответов нет</p>	Е
7.	<p>Как называется именованная спецификация одного или более столбцов (для каждого столбца указывается имя, а также его тип или домен)? (Отметьте один правильный вариант ответа.) A) строчный тип данных B) объектный тип данных C) комбинированный тип данных D) правильных ответов нет</p>	А

8.	<p>Что понимается под наследованием типов? (Отметьте один правильный вариант ответа.)</p> <p>А) однородная масса разрядов, имеющая какую либо структуру В) возможность дисциплинированного создания новых типов на основе уже определенных С) многоходовой программный модуль, точки входа которого соответствуют набору операций реализуемого типа D) правильных ответов нет</p>	В
9.	<p>Что из перечисленного не относится к наиболее распространенным конструируемым типам данных? (Отметьте один правильный вариант ответа.)</p> <p>А) тип записи В) тип множества С) тип массива D) тип распределения E) правильных ответов нет</p>	D
10.	<p>Что представляет собой открытый массив? (Отметьте один правильный вариант ответа.)</p> <p>А) фактический параметр подпрограммы, описывающий базовый тип элементов массива, но не определяющий его размерности и границы В) формальный параметр подпрограммы, описывающий базовый тип элементов массива и определяющий его размерность и границы С) формальный параметр подпрограммы, описывающий базовый тип элементов массива, но не определяющий его размерности и границы D) правильных ответов нет</p>	С
11.	<p>Каким способом параметр массив всегда передается в функцию? (Отметьте один правильный вариант ответа.)</p> <p>А) по значению В) по направлению С) по адресу D) правильных ответов нет</p>	С
12.	<p>Что понимается под классом в объектно-ориентированном программировании (ООП)? (Отметьте один правильный вариант ответа.)</p> <p>А) процедуры и функции любого языка программирования В) структурный тип данных, который включает описание полей данных, процедур и функций, работающих с этими полями данных С) списки передаваемых параметров, типы функции D) правильных ответов нет</p>	В
13.	<p>Какие синтаксические правила объявления переменных являются корректными:</p> <p>А) <тип> [<атрибуты>] [<модификаторы>] <список объявителей>; В) <список объявителей> : <тип>; С) [<атрибуты>] [<модификаторы>] <тип> <список объявителей>; D) [<атрибуты>] [<модификаторы>] <тип> : <список объявителей>; E) правильных ответов нет</p>	С
14.	<p>Какие типы в языке С# относятся к значимым: (Ответ считается верным, если отмечены все правильные варианты ответов)</p> <p>А) логический В) арифметический С) структуры</p>	А, В, С, D

	D) перечисление E) массивы F) строки G) классы H) правильных ответов нет	
15.	Определение класса в C# и других объектных языках содержит: (Ответ считается верным, если отмечены все правильные варианты ответов) A) вершины B) методы C) приключения D) равнины E) поля F) события G) правильных ответов нет	B, E, F
16.	Какие типы в языке C# относятся к ссылочным: (Ответ считается верным, если отмечены все правильные варианты ответов) A) логический B) арифметический C) структуры D) перечисление E) массивы F) строки G) классы H) правильных ответов нет	E, F, G
17.	Какой из перечисленных типов языка C# не относится к значимым: (Отметьте один правильный вариант ответа.) A) тип double B) логический тип C) тип, задающий структуру D) тип, задающий массив E) все арифметические типы F) текстового документа.	D
18.	Отметьте истинные высказывания: (Ответ считается верным, если отмечены все правильные варианты ответов) A) память, отводимая переменной значимого типа, принадлежит переменной и ни с какой другой переменной не разделяется B) память, отводимая переменной ссылочного типа, принадлежит переменной и ни с какой другой переменной не разделяется C) значения переменных значимого типа нельзя изменять D) для ссылочных типов значение задается ссылкой на некоторый объект, расположенный в динамической памяти - "куче" E) несколько ссылочных переменных могут ссылаться на один и тот же объект и разделять его значения F) правильных ответов нет	A, D, E

№ п/п	Задание Блока Б (вопрос)	Эталон ответа
Инструкция по выполнению заданий № 19-25: В соответствующую строку бланка ответов запишите ответ выполненного задания, окончание предложения или пропущенные слова		

19.	Перевод программы с языка программирования на язык машинных кодов называется ...	трансляция
20.	Процесс исправления ошибок в программе, при котором цель исправить все ошибки не ставится, называется.	отладка
21.	Процесс, посредством которого проверяется правильность программы называется .	тестирование
22.	Выражения строятся из операндов - ,, ,, - объединенных знаками операций и скобками	констант, переменных, функций
23.	Под перегрузкой операции понимается существование нескольких реализаций одной и той же .	операции
24.	Метод класса называется . , если существует несколько реализаций этого метода.	перегруженным
25.	Массивом называют упорядоченную совокупность элементов . типа.	одного
19.	Перевод программы с языка программирования на язык машинных кодов называется .	трансляция
20.	Процесс исправления ошибок в программе, при котором цель исправить все ошибки не ставится, называется.	отладка
21.	Процесс, посредством которого проверяется правильность программы называется .	тестирование

Вопросы для подготовки к текущему контролю

1. Этапы решения задач на компьютере.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
4. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
5. Интегрированная среда разработки приложений.
6. Структура программы, элементы языка (алфавит). Понятие типа данных.
7. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
8. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
9. Условный оператор. Оператор множественного ветвления (выбора).
10. Циклы: с предусловием, с постусловием. Связь с другими циклами.
11. Циклы: с параметром. Связь с другими циклами.
12. Структурированные типы данных.
13. Сортировка массивов.
14. Подпрограммы.
15. Процедуры и функции.
16. Простые типы данных.
17. Множества.
18. Комбинированный тип данных (записи).
19. Файловые типы.
20. Текстовые файлы.
21. Прямой и последовательный доступ к компонентам файла.
22. Поиск в типизированных файлах
23. Типизированные файлы.
24. Статическая и динамически распределяемая память.
25. Динамические структуры данных.
26. Модуль. Общая структура модуля. Компиляция и подключение модуля.
27. Объектно-ориентированное программирование. Компоненты.

Примеры практико-ориентированных заданий для подготовки к текущему контролю

1. Подсчитать сумму элементов над главной диагональю квадратной матрицы и под ней.
2. Определить максимальный и минимальный элементы двумерного массива и их индексы.
3. Заполнить одномерный массив случайными неповторяющимися числами.
4. Написать программу, реализующую игру «Угадай число».
5. Написать программу решения квадратного уравнения.
6. Расположить строки двумерного массива в порядке возрастания сумм элементов в них.
7. Заполнить одномерный массив случайными числами из интервала $[-10, 10]$. Упорядочить массив по возрастанию, если сумма элементов массива положительна, или по убыванию в противном случае.
8. Сохранить данные одномерного массива в файле данных. Организовать чтение данных из файла.
9. Решить линейное неравенство, не используя условную конструкцию.

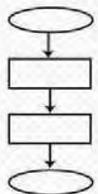
10. Определить количество слов во введенном тексте.
11. Заполнить двумерный массив случайными числами. Вывести массив на экран. Найти максимальный и минимальный элементы, их индексы.
12. Заполнить одномерный массив случайными числами. Упорядочить массив по возрастанию или убыванию по желанию пользователя.
13. Подсчитать, сколько среди пятизначных чисел таких, у которых сумма четных цифр равна сумме нечетных цифр.
14. Определить, сколько раз в тексте встречается указанный фрагмент.
15. Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.
16. Создать проект, позволяющий сохранять текст, набранный в текстовом окне в файле данных, очищать текстовое окно и загружать в него сохраненный текст.
17. Создать проект, позволяющий построить треугольник по координатам вершин.
18. Имеется десять флажков. Создать проект, позволяющий сохранять состояние флажков, обнулять и восстанавливать сохраненное состояние.
19. Организовать анимацию процесса движения шарика по окружности.
20. Создать проект, реализующий часы с «будильником».
21. Проект содержит три текстовых окна. Используя буфер обмена, организовать обмен информацией между приложениями.
22. Вывести алфавит введенного пользователем текста.
23. Создать проект, позволяющий добавлять записи в простой список и удалять их.
24. Расположить слова строкового массива в порядке возрастания их длин.
25. Построить на экране правильный N-угольник. Количество вершин и радиус окружности, на которой они расположены, указываются пользователем.
26. Построить график функции в выбранной пользователем системе координат.
27. Составить программу, генерирующую цвет по трем его составляющим (красный, зеленый, синий).
28. Составить программу, моделирующую работу светофора.
29. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран в любую его точку правильный многоугольник произвольного размера с произвольным количеством вершин.
30. Написать программу, моделирующую движение броуновской частицы.
31. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран график функции. График должен выводиться в прямоугольную область указанного размера. Объект должен иметь свойство, определяющее величину единичного отрезка.
32. Разработать класс, объекты которого фиксируют время своего существования.
33. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран анимацию вращающегося шарика. Координата точки - оси вращения, скорость вращения, радиус вращения и направление вращения задаются.
34. Описать процедуру, имеющую три аргумента целого типа и работающую по следующему принципу: значения трех переменных целого типа, описанных выше точки вызова и указанных в качестве аргументов процедуры, должны быть расположены по возрастанию в порядке указания аргументов в процедуре.
35. Разработать класс, объекты которого обладают методами, позволяющими заполнить одномерный массив случайными неповторяющимися числами, вывести его на экран и упорядочить.
36. Решить квадратное уравнение, используя директивы препроцессора и условную операцию.
37. Описать процедуру, позволяющую произвести обмен значениями переменных указанных в качестве аргумента.
38. Разработать компонент, предлагающий интерфейс для отображения одномерного массива и имеющий методы для заполнения, очистки и сортировки.

39. Разработать компонент «Переключатель», имеющий пять стационарных состояний. Компонент должен иметь свойство для установки состояния и обрабатывать событие, связанное с выбором другого состояния.
40. На форме имеется десять командных кнопок (пронумерованных от 1 до 10) и графический контейнер. При нажатии на одну из кнопок в контейнере должно появиться соответствующее римское число.
41. Написать программу, позволяющую просматривать записи простейшей базы данных, добавлять, корректировать и удалять их.
42. Разработать программу в Lazarus, позволяющую заполнять двумерный массив с указанным пользователем количеством строк и столбцов и находить среднее арифметическое в каждой строке и в каждом столбце.
43. Используя общий диалог, написать программу, позволяющую сохранять указанный пользователем текст в файле и считывать сохраненный текст из файла.
44. Используя объектно-ориентированный подход, написать программу, моделирующую движение броуновских частиц.
45. Разработать компонент «Таймер», выводящий время, прошедшее с момента запуска таймера. Компонент должен иметь метод, возвращающий текущее время, обрабатывать событие, связанное с включением и выключением таймера и метод, обнуляющий таймер.
46. Описать класс, объекты которого связываются с текстовыми окнами, хранят введенную пользователем в окно информацию, обладают методом сортировки символов в текстовых окнах в алфавитном порядке и методом восстанавливающим текст в окне после сортировки.
47. Разработать компонент, предлагающий интерфейс для ввода данных и вывода решения квадратного уравнения.
48. Создать проект, моделирующий работу секундомера. Секундомер должен иметь циферблат с минутной и секундной стрелками, метку и одну командную кнопку.
49. Необходимо написать консольную программу, где пользователь будет вводить с клавиатуры 2 числа. Числа будут сравниваться с последующим выводом в консоль результата этого сравнения (равны ли значения, а если нет, то какое число больше/меньше).
50. Пусть пользователь имеет возможность ввести одно число, причем: когда значение больше пяти и одновременно с этим меньше десяти, выводится один текст с соответствующим сообщением; когда значение не попадает в указанный диапазон, программа его не узнает, выводя другой текст с соответствующим сообщением.

Раздел 4. Язык программирования Python

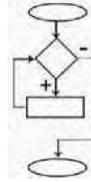
Тестовые задания для проведения текущего контроля и подготовки к промежуточной аттестации

1. Язык программирования Python подходит для разработки:
 - a. Компьютерных и мобильных приложений
 - b. Аналитика и машинное обучение
 - c. Игр
 - d. Ничего из этого.
2. Назовите тип алгоритма:



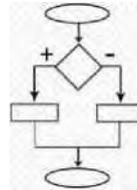
- a. Разветвляющийся
- b. Линейный
- c. Циклический
- d. Смешанный

3. Назовите тип алгоритма:



- a. Линейный
- b. Разветвляющийся
- c. Циклический
- d. Смешанный

4. Назовите тип алгоритма:



- a. Разветвляющийся
- b. Линейный
- c. Смешанный
- d. Циклический

5. Что хранит в себе переменная?

- a. Имя
- b. Значение
- c. Тип
- d. Длину своего значения

6. Что обозначает тип данных `int`?

- a. Целочисленное
- b. Вещественное
- c. Строковое
- d. Булево

7. Выберите правильную запись оператора присваивания:

- a. `10 = x`
- b. `y = 7,8`
- c. `a = 5`
- d. `a == b + x`

8. Укажите оператор ввода:

- a. `input()`
- b. `print()`
- c. `int()` d. `random()`

9. Сколько возможных значений у переменной типа `bool`? a. 2 b. 4 c. 10

- d. Сколько угодно

10. Какой оператор здесь используется?

If `n < 100`: `b = n + a` a. Условный оператор b. Оператор присваивания c. Оператор сложения d. Оператор умножения

11. Что лучше использовать для множественного ветвления? a. `if - elif - else` b. Много `if` c. `if - else - elif` d. `while`

12. Оператор цикла в языке Python: a. `while` b. `for` c. `if` d. `print`

13. Сколько раз произойдет итерация цикла? (Итерация - единичное выполнение тела цикла)
- `print("Осталось", total)`
- ```
total = 100
i = 0
while i < 0:
 n = int(input())
 total = total - n
 i = i + 1
```
- a. 4  
b. 5  
c. 6 d. 0
14. Для чего нужен оператор `break`?  
a. Для завершения программы  
b. Для выхода из цикла  
c. Для поломки компьютера  
d. Для удаления программы
15. Где находятся параметры, а где аргументы функции?  
a. Параметры пишутся при объявлении функции, аргументы при вызове  
b. Аргументы пишутся при объявлении функции, параметры при вызове  
c. Это одно и то же!  
d. У функции есть только параметры
16. Что делает функция `len()`?  
a. Возвращает длину строки  
b. Возвращает случайное число  
c. Возвращает номер символа  
d. Возвращает модуль числа
17. Как добавить модуль в программу?  
a. `import math`  
b. `import math()`  
c. `import (math)`  
d. `import.math`
18. На каких операционных системах может работать Python?  
a. Windows  
b. Linux  
c. macOS  
d. Ничего из этого
19. От чего язык программирования называется «Питон»?  
a. В честь змеи  
b. В честь ТВ-шоу  
c. В честь игры  
d. В честь блюда
20. Сколько уже лет языку программирования Python?  
a. 30  
b. 19  
c. 7  
d. 15
21. Создатель языка программирования Python  
a. Гвидо Ван Россум  
b. Дэвид Паттерсон  
c. Эрвин Дональд Кнут  
d. Джеймс Артур Гослинг
22. `a = 345`. Что выведет команда `print(a//100)`  
a. 3  
b. 5  
c. 4  
d. 34
23. Выберите циклический алгоритм

- a. `k = 0`  
`while k < 10:`  
`print("npueem")`  
`k += 1`
- b. `a = int(input())`  
`b = int(input())`  
`c = int(input())`  
`s = a+b+c`  
`print(c)`
- c. `a = int(input())`  
`if a > 0:`  
`print(a)`  
`else:`  
`print(a)`

24. В какой строке правильно записан ввод числа с клавиатуры?

|                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>1. <code>a = int(input)</code></li> <li>2. <code>b=input(int())</code></li> <li>3. <code>c=int(input())</code></li> <li>4. <code>s=a+b+c</code></li> <li>5. <code>print(s)</code></li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- a. 1
- b. 2
- c. 3
- d. 4
- e. 5

25. В какой строке допущена ошибка?

|                                                                                                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> <li>1. <code>a =</code><br/> <code>    int(input())</code></li> <li>2. <code>if a&gt;0:</code></li> <li>3. <code>print(a):</code></li> <li>4. <code>else</code></li> <li>5. <code>print(A)</code></li> </ul> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- a. 1
- b. 2
- c. 5
- d. 4

26. Что такое «else»?

- a. Так как
- b. Иначе
- c. Если
- d. Потому что

27. Сколько раз программа напишет слово «Пока»?

|                                                         |
|---------------------------------------------------------|
| <pre> k=0 while k&lt;10:  print(«npueem») k += 1 </pre> |
|---------------------------------------------------------|

- a. 9
- b. 0
- c. 10

d. Бесконечно

**Примеры практико-ориентированных заданий для текущего контроля**

1. Как получить список всех атрибутов объекта?
2. Как получить список всех публичных атрибутов объекта?
3. Как получить список методов объекта?
4. В какой "магической" переменной хранится содержимое `help`?
5. Есть два кортежа, получить третий как конкатенацию первых двух
6. Есть два кортежа, получить третий как объединение уникальных элементов первых двух кортежей
7. Почему если в цикле меняется список, то используется `for x in lst[:]`, что означает `[:]`?
8. Есть два списка одинаковой длины, в одном ключи, в другом значения. Составить словарь.
9. Есть два списка разной длины, в одном ключи, в другом значения. Составить словарь. Для ключей, для которых нет значений использовать `None` в качестве значения. Значения, для которых нет ключей игнорировать.
10. Есть словарь. Инвертировать его. Т.е. пары ключ: значение поменять местами — значение: ключ.
11. Есть строка в юникоде, получить 8-битную строку в кодировке `utf-8` и `cp1251`
12. Есть строка в кодировке `cp1251`, получить юникодную строку

### 3. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

#### МДК 01.01. Разработка программных модулей

Форма промежуточной аттестации — экзамену

##### Вопросы к экзамену

1. Особенности создания программного продукта. Проблематика проектирования.
2. Понятие технологии разработки программы. Модели жизненного цикла.
3. Унифицированный язык моделирования (UML). Определение прецедентов (вариантов использования).
4. Определение функциональных и эксплуатационных требований к программным продуктам при структурном и объектном подходе.
5. Словарь терминов. Функциональные диаграммы. Диаграммы потоков данных (DFD).
6. Проектирование программного обеспечения при структурном подходе.
7. Структурная схема разрабатываемого программного обеспечения (логическая и физическая модели)
8. Метод пошаговой детализации при составлении алгоритмов.
9. Пользовательские типы данных: структуры (записи), объединения, перечисления, множества.
10. Синтаксис объявления пользовательских типов данных. Доступ к элементу пользовательского типа.
11. Основы файловой системы. Порядок работы с текстовыми и двоичными файлами.
12. Функции для работы с файлами разного типа. Правила бесформатного и форматного ввода/вывода.
13. Разделение программы на подпрограммы. Параметры подпрограммы. Перегрузка подпрограмм.
14. Группировка подпрограмм и связанных с ними данных в отдельные файлы - модули. Основные характеристики программного модуля.
15. Модульная структура программных продуктов.
16. Проектирование программного обеспечения при объектном подходе.
17. Структура проекта в объектно-ориентированной среде программирования. Событийно-управляемая модель программирования.
18. Компоненты объектно-ориентированной среды. Принципы разработки графического интерфейса.
19. Особенности ввода/вывода в объектно-ориентированной среде программирования.
20. Классы. Конструкторы и деструкторы. Наследование. Создание наследованного класса.
21. Статический и динамический полиморфизм (виртуализация методов). Дружественные функции и классы.
22. Механизм обработки и генерирования исключительных ситуаций. Способы создания DLL-библиотеки.
23. Серия стандартов ISO 9000. Процесс сертификации программ на базе информации об их использовании.
24. Виды программных документов. Пояснительная записка. Руководство пользователя. Руководство системного программиста.
25. Тестирование методами «белого ящика» и «черного ящика».
26. Порядок разработки тестов. Автоматизация тестирования.
27. Модульное, интеграционное, системное тестирование.
28. Отладка программ. Автономная и комплексная отладка программ.

29. Инструментальные средства отладки программ.
30. Основные принципы построения приложений, основанных на СУБД.
31. Компоненты объектно-ориентированной среды, необходимые для организации базы данных и работы с ней.
32. Принцип разработки программы, управляющей локальной базой данных. Способы навигации по набору данных.
33. Вставка, добавление и удаление записей таблиц базы данных. Поиск, фильтрация и сортировка данных. Возможности среды для создания новых полей.
34. Принцип разработки программы, управляющей удаленной базой данных. SQL-запросы на выборку, добавление, редактирование и удаление данных. Агрегатные функции. Группировка записей.
35. Статические и динамические параметрические SQL-запросы.

#### **Критерии оценки**

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки

**МДК 01.02. Поддержка и тестирование программных модулей** Форма промежуточной аттестации — **дифференцированный зачет.**

#### **Вопросы к дифференцированному зачету**

1. Введение: тестирование - способ обеспечения качества программного продукта
2. Основные понятия тестирования
3. Подходы к обоснованию истинности формул и программ и их связь с тестированием. Вопросы организации тестирования. Фазы тестирования, основные проблемы тестирования и поставлена задача выбора конечного набора тестов.
4. Требования к идеальному критерию тестирования и классы частных критериев. Особенности применения структурных и функциональных критериев на базе конкретных примеров. Особенности применения методов стохастического тестирования и метод оценки скорости выявления ошибок. Мутационный критерий и на примере иллюстрируется техника работы с ним.
5. Оценка оттестированности проекта: метрики и методика интегральной оценки
6. Графовые модели проекта, метрики оценки оттестированности проекта, приводятся примеры плоской и иерархической моделей проекта.
7. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных, динамические и статические методы при структурном подходе. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования. Рассматриваются особенности интеграционного тестирования в процедурном программировании.

8. Интеграционное тестирование и его особенности для объектно-ориентированного программирования
9. Модель объектно-ориентированной программы. Оценки сложности тестирования и методика тестирования объектно-ориентированной программы. Рассматривается пример интеграционного тестирования.
10. Разновидности тестирования: системное и регрессионное тестирование
11. Автоматизация тестирования
12. Особенности индустриального тестирования
13. Документирование и оценка индустриального тестирования
14. Описываются особенности документирования тестовых процедур для ручных и автоматизированных тестов, описаний тестовых наборов и тестовых отчетов. Рассматривается жизненный цикл дефекта. Обсуждаются метрики, используемые при тестировании.
15. Регрессионное тестирование: цели и задачи, условия применения, классификация тестов и методов отбора
16. Регрессионное тестирование: разновидности метода отбора тестов
17. Регрессионное тестирование: методики, не связанные с отбором тестов и методики порождения тестов
18. Регрессионное тестирование: алгоритм и программная система поддержки
19. Описание тестируемой системы и ее окружения. Планирование тестирования
20. Модульное тестирование на примере классов
21. Интеграционное тестирование
22. Системное тестирование
23. Ручное тестирование
24. Автоматизация тестирования с помощью скриптов
25. Автоматическая генерация тестов на основе формального описания
26. Описание ручного тестирования
27. Автоматизация тестирования с помощью скриптов
28. Описание автоматической генерации MSC тестов

#### **Критерии оценки (недифференцированной)**

Оценка «зачтено» выставляется студенту, проявившему знания основного программного материала в объеме, необходимом для последующего обучения и предстоящей практической деятельности, знакомому с основной рекомендованной литературой, но допустившему неполные или слабо аргументированные ответы, испытывающему затруднения.

Оценка «не зачтено» выставляется студенту, обнаружившему значительные пробелы в знании программного материала, допустившему принципиальные ошибки при применении теоретических знаний, которые не позволяют ему продолжить обучение или приступить к практической деятельности без дополнительной подготовки по данной дисциплине.

#### **МДК 01.03. Разработка мобильных приложений**

Форма промежуточной аттестации — экзамен.

##### **Вопросы к экзамену**

1. История возникновения мобильных операционных систем
2. Основные этапы становления рынка мобильных приложений
3. Современное состояние рынка мобильных приложений
4. Классификация видов мобильных приложений
5. Преимущества использования мобильных приложений в сравнении с веб-приложениями
6. Недостатки использования мобильных приложений в сравнении с десктопными приложениями

7. Инструментальные среды разработки мобильных приложений для операционной системы Apple iOS
8. Инструментальные среды разработки мобильных приложений для операционной системы Android
9. Инструментальные среды разработки мобильных приложений для операционной системы Windows Phone
10. Структура операционной системы iOS
11. Структура операционной системы Android
12. Структура приложения iOS
13. Структура приложения Android
14. Основные требования к интерфейсу приложений iOS
15. Основные требования к интерфейсу приложений Android
16. Android-манифест
17. Взаимодействие Android-приложения с сетью
18. Работа Android-приложения с локальной базой данных
19. Считывание информации Android-приложением с XML-файла
20. Вызов приложения из другого приложения в ОС Android
21. Проблемы безопасности мобильных операционных систем
22. Бизнес-модели распространения мобильных приложений
23. Стратегия размещения приложения на Google Play
24. Стратегия размещения приложения на AppStore
25. Сравнительная характеристика современных мобильных операционных систем
26. HTML5 и мобильные приложения
27. Проблемы совместимости мобильных приложений со старыми версиями операционных систем
28. Проектирование мобильных приложений с использованием C++
29. Технологии фреймворков в проектировании мобильных приложений
30. Проблемы масштабирования СУБД в мобильных приложениях
31. Основные технологии виртуализации в инструментальных средах при создании мобильных приложений
32. Проблемы обеспечения безопасности в платных мобильных приложениях
33. Перспективы развития рынка мобильных приложений в России

### **Критерии оценки**

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки

### **МДК 01.04. Системное программирование**

Форма промежуточной аттестации — экзамен.

## Вопросы к экзамену

1. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры.
2. Дайте характеристику языку ассемблера, определение, преимущества.
3. Двумерные массивы (списки списков). Вложенная генерация.
4. Для чего используются трансляторы, компиляторы, интерпретаторы.
5. Интегрированная среда разработки приложений.
6. Классы в Python. Конструктор класса.
7. Классы в Python. Наследование.
8. Классы в Python. Перегрузка операторов.
9. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. C#
10. Кортежи в Python. Создание и использование. Кортежи переменных
11. Литералы чисел и строк в Python
12. Модуль. Общая структура модуля. Компиляция и подключение модуля. C#
13. Назовите и опишите константы, метки и условную компиляцию.
14. Область действия переменных и других идентификаторов.
15. Объектно-ориентированное программирование. Компоненты.
16. Объясните двоичное кодирование информации. Представление элементарных типов данных: натуральные числа, целые числа со знаком, числа с плавающей точкой.
17. Регистры eax, ebx, ecx, edx и их специальные свойства.
18. Операции (арифметические, логические) на типах. Стандартные функции. Выражения. C#
19. Опишите назначение простейших команд Ассемблера. Расскажите про команды ADC и SBB.
20. Параметры-переменные, параметры-значения, параметры-константы. Примеры.
21. Перечислите и опишите этапы проектирования и выполнения программ
22. Перечислите и охарактеризуйте средства аппаратной поддержки функций ОС.
23. Подпрограммы. Основные способы передачи параметров в подпрограмму, их сравнение. C#
24. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма. Понятие исполнителя. Система команд исполнителя.
25. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
26. Понятие процесса и потока: раскройте и охарактеризуйте.
27. Принципы структурного программирования. Блочный (модульный) подход к построению алгоритмов решения задач.
28. Простые типы данных. Структурированные типы данных. Строковый тип данных, примеры. C#
29. Процедуры, ориентированные на работу со строками. Примеры.
30. Процедуры. Вызов процедуры в программах. Формальные и фактические параметры. Соответствие между параметрами. Примеры.
31. Процедуры. Организация и вызов. Примеры. C#
32. Расскажите как формируются предложения на Ассемблере из лексем. Опишите операнды Ассемблера
33. Расскажите про команды MUL, IDIV, IMUL и DIV.
34. Расскажите про компилятор ассемблер - программ, редактор связей (загрузчик).
35. Расскажите про обработку прерываний. Контекст процесса (потока). Особенности операционных систем реального времени.
36. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.

37. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.
38. Сортировка массивов. C#
39. Списки в Python. List comprehensions: генерация списков.
40. Списки в Python. Методы списков и операции со списками
41. Срезы списков. Присваивание в срез. Проблема копирования списка
42. Строки в Python. Методы find, count, replace, strip, upper, lower и другие.
43. Строки в Python. Срезы с двумя и тремя параметрами
44. Структура программы, элементы языка (алфавит). Понятие типа данных. C#
45. Структурированные типы данных. Линейные массивы. Примеры задач. Двумерные массивы. Примеры задач. C#
46. Условный оператор и каскадная условная конструкция в Python
47. Условный оператор. Оператор множественного ветвления (выбора). C#
48. Файловые типы. Общие процедуры для работы с файлами. Компонентные (типизированные) файлы. C#
49. Функции, ориентированные на работу со строками. Примеры.
50. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
51. Функции. Организация и вызов. Примеры. C#
52. Цикл for и его особенности в Python. Функция range
53. Циклы: с параметром. Связь с другими циклами. C#
54. Циклы: с предусловием, с постусловием. Связь с другими циклами. C#
55. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).

#### **Критерии оценки**

«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.

«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.

«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.

