**ОПК-7. Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Индикаторы, дескрипторы  достижения компетенций | | Комплекс заданий для оценки компетенций | Ключи правильных ответов | Критерии оценки  в баллах | | |
| **Дисциплина «Языки программирования» (Б1.Б.26)** | | | | | | |
| **ИОПК-7.1.**  Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения.  **Знать:**  этапы решения задач на ЭВМ, основные конструкции языков высокого уровня, структурные типы данных, основы алгоритмизации;  приемы и методы разработки программ с применением процедурного подхода, его достоинства и недостатки;  базовые особенности реализации принципов структурного и объектно-ориентированного программирования.  **Уметь**:  оптимизировать алгоритмы решения задач, выбирать структуры данных для реализации решений ;  проектировать и реализовывать программы на языках Pascal и С++ в средах разработки Borland Pascal и Microsoft Visial Studio C++, анализировать полученные результаты .  **Владеть**:  - практическими навыками написания, отладки и тестирования программного продукта средствами изученных интегрированных сред разработки | **1.** Выберите правильный вариант ответа.  Что из перечисленного не является типом данных в Java: **int; float; real** | | | | **real** | Верный ответ – 1 неверный – 0. |
| **2.** Выберите правильный вариант ответа.  Какой из перечисленных далее классов представляет собой коллекцию, построенную на основе массива?  Укажите нужную букву.  А. LinkedList  Б. ArrayList  В. AbstractCollection | | | | Б | Верный ответ – 1 неверный – 0. |
| **3.** Выберите и запишите правильный вариант ответа.  Какой метод используется для запуска потока в Java; **run(); start(); execute()** | | | | **start()** | Верный ответ –1 неверный – 0. |
| **4.** Выберите и запишите правильный вариант ответа.  Какое ключевое слово используется для создания неизменяемой переменной: **final; static; const** | | | | **final** | Верный ответ – неверный – 0. |
| **5.**Установите соответствие:   |  |  |  |  | | --- | --- | --- | --- | | А | Объединение данных и методов, работающих с ними; обеспечение публичного интерфейса, скрытие реализации. | 1 | Абстрагирование | | Б | Это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. | 2 | Инкапсуляция | | В | Описание нового класса на основе уже существующего с частично или полностью заимствующейся функциональностью | 3 | Наследование | | Г | Использование объектов с одинаковыми интерфейсами без информации о типе и внутренней структуре объектов. | 4. | Полиморфизм | | | | | |  |  | | --- | --- | | А | 2 | | Б | 1 | | В | 3 | | Г | 4 | | Верный ответ –1 неверный – 0. |
| **6.** Установите соответствие:   |  |  |  |  | | --- | --- | --- | --- | | А | «Класс-заготовка», на основе которой с помощью наследования позже  будут созданы конкретные классы. | 1 | Интерфейс | | Б | Абстрактный тип, содержащий объявления методов и констант. | 2 | Класс | | В | Шаблон или модель, по которой создаются объекты. | 3 | Абстрактный класс | | | | | |  |  | | --- | --- | | А | 3 | | Б | 1 | | В | 2 | | Верный ответ –1 неверный – 0. |
| **7.** Установите соответствие:   |  |  |  |  | | --- | --- | --- | --- | | А | Части кода, помеченные данным модификатором, предназначаются для конечного пользователя. | 1 | Private | | Б | Ограничивает видимость данных и методов пределами одного класса. | 2 | Protected | | В | Поля и методы будут видны в пределах всех классов, находящихся в том же пакете, и в классах-наследниках. | 3 | Public | | | | | |  |  | | --- | --- | | А | 3 | | Б | 1 | | В | 2 | | Верный ответ –1 неверный – 0. |
| **8.** Установите соответствие:   |  |  |  |  | | --- | --- | --- | --- | | А | Ассоциативный массив | 1 | List | | Б | Коллекция без дубликатов | 2 | Set | | В | Упорядоченный список с позиционным доступом | 3 | Queue | | Г | Очередь элементов или стэк | 4 | Map | | | | | |  |  | | --- | --- | | А | 4 | | Б | 2 | | В | 1 | | Г | 3 | | Верный ответ –1 неверный – 0. |
| **9.** Установите правильную последовательность работы метода **equals()** в Java:  1) сравнение значений полей двух объектов,  2) проверка на равенство ссылок на объекты,  3) возвращение true, если объекты равны, иначе false. | | | | 2, 1, 3 | Верный ответ –1 неверный – 0. |
| **10.** **Установите правильную последовательность объявления метода в Java:**  1) фигурные скобки {}  2) имя метода  3) список параметров в круглых скобках ()  4) тип возвращаемого значения | | | | 4, 2, 3, 1 | Верный ответ –1 неверный – 0. |
| **11.** Установите правильную последовательность объявления цикла **for:**  1) ключевое слово for  2) круглые скобки ()  3) фигурные скобки {}  4) инициализация счетчика,  5) условие,  6) шаг изменения счетчика. | | | | 1, 2, 4, 5, 6, 3 | Верный ответ – неверный – 0. |
| **12.** Установите правильную последовательность объявления массива, инициализированного **с помощью фигурных скобок:**  1) квадратные скобки,  2) тип данных,  3) имя массива,  4) элементы массива,  5) фигурные скобки {} | | | | 2, 1, 3, 5, 4 | Верный ответ –1 неверный – 0. |
| **13.** Введите ответ в поле ввода (в именительном падеже с маленькой буквы)  Конструкторы в Java имеют то же самое имя, что и \_\_\_\_\_\_. | | | | класс | Верный ответ – 1 неверный – 0. |
| **14.** Введите ответ в поле ввода (в именительном падеже с маленькой буквы)  Ключевое слово \_\_\_\_\_\_\_ используется для вызова конструктора родительского класса. | | | | super | Верный ответ – 1 неверный – 0. |
| **15.** Введите ответ в поле ввода (в именительном падеже с маленькой буквы)  Для создания экземпляра класса используется оператор \_\_\_\_\_\_\_\_. | | | | new | Верный ответ – 1 неверный – 0. |
| **Дисциплина «Технологии и методы программирования» (Б1.Б.27)** | | | | | | |
| **ИОПК -7.1.**  Применяет современные языки программирования, технологии и инструменты для создания програм-много обеспечения (ПО)  **ИОПК -7.2.**  Выполняет выбор инструментов и технологий для обеспечения качества программного обеспечения.  **Знать:**  –тенденции развития современных технологий программирования и ПО (ИОПК-7.1, 7.2).  **Уметь:**  строить современные автоматизированные системы обработки информации (ИОПК -7.1, 7.2).  **Владеть:**  инструментами по разработке программного обеспечения (ИОПК-7.1, 7.2). | 1. Какой из процессов не относится к процессам жизненного цикла Программного обеспечения (ПО):   Основной,вспомогательный, организационный, технический | | | | технический | Верный ответ – 1 неверный – 0. |
| 1. Укажите цифру неправильного варианта тестирования ПО по знанию о системе:   1) тестирование чёрного ящика,  2) тестирование белого ящика,  3) тестирование красного ящика. | | | | 3 | Верный ответ – 1 неверный – 0. |
| 3. Выберите атрибуты, относящиеся к группе атрибутов качества ПО «эффективность»:  1) временные характеристики,  2) ресурсные характеристики,  3) характеристики защищенности. | | | | 1, 2 | Верный ответ – 1 неверный – 0. |
| 1. **Установите соответствие между названием и определением метода тестирования ПО**:  |  |  |  |  | | --- | --- | --- | --- | | А | Отладка | 1 | попытка найти ошибки при выполнении программы в реальной среде | | Б | Контроль | 2 | поиск места в программном коде, где расположена ошибка | | В | Испытание | 3 | поиск ошибок при выполнении программ в тестовой или моделируемой среде | | | | | |  |  | | --- | --- | | А | 2 | | Б | 3 | | В | 1 | | Верный ответ – 1 неверный – 0. |
| 1. **Установите соответствие между названием и определением метода отладки ПО**  |  |  |  |  | | --- | --- | --- | --- | | **А** | Научный метод отладки | **1** | строим гипотезу и её доказываем, как правило, знание об исследуемой системе низкое | | **Б** | Отладка с помощью интуиции | **2** | это способность, взглянув на симптомы, быстро сузить область поиска до места кода, в котором проблема возникла, как правило, знание о системе высокое | | **В** | Внезапное озарение | **3** | всплеск понимания, где расположен дефект кода, как правило,  характеризуется средним знанием о системе | | | | | |  |  | | --- | --- | | А | 1 | | Б | 2 | | В | 3 | | Верный ответ – 1 неверный – 0. |
| 6 **Установите соответствие между названием и содержимым программного документа:**   |  |  |  |  | | --- | --- | --- | --- | | А | Руководство администратора | 1 | Назначение и возможности программы, пользователя, порядок решения типовых задач, описание функций программы | | Б | Руководство пользователя | 2 | Архитектура программы или создаваемых на ее основе приложений, описание программных интерфейсов к ее объектам, протоколов обмена данными и т.п | | В | Руководство программиста | 3 | Управление учетными записями пользователей, назначение пользователям системы, обеспечивающей ее целевое применение и прав доступа, ведение нормативно-справочной информации, загрузка и выгрузка данных | | Г | Руководство системного  администратора | 4 | Установка программы, ее интеграция в систему, проверка правильности функционирования, устранение аварийных ситуаций | | | | | |  |  | | --- | --- | | А | 3 | | Б | 1 | | В | 2 | | Г | 4 | | Верный ответ – 1 неверный – 0. |
| 1. Установите соответствие отдельных атрибутов определенной группе атрибутов качества ПО:  |  |  |  |  | | --- | --- | --- | --- | | А | Группа Эффективность | 1 | Временная эффективность | | Б | Группа Сопровождаемость | 2 | Удобство внесения изменений | | В | Группа Переносимость | 3 | Адаптируемость | | | | | |  |  | | --- | --- | | А | 1 | | Б | 2 | | В | 3 | | Верный ответ – 1 неверный – 0. |
| 1. Установите соответствие между названием и определением:  |  |  |  |  | | --- | --- | --- | --- | | А | Бизнес-требования | 1 | описывают свойства системы (удобство использования, безопасность, надёжность, расширяемость и т.д.), которыми она должна обладать при реализации своего поведения | | Б | Пользовательские требования | 2 | описывают, почему организации нужна такая система | | В | Нефункциональные требования | 3 | описывают цели или задачи, которые пользователи должны иметь возможность выполнять с помощью системы | | | | | |  |  | | --- | --- | | А | 2 | | Б | 3 | | В | 1 | | Верный ответ – 1 неверный – 0. |
| **9.** Установите правильную последовательность **процесса тестирования**:  1) сравниваем ожидаемый и фактический результаты;  2) узнаем ожидаемый результат;  3) узнаем фактический результат. | | | | 2, 3, 1 | Верный ответ – 1 неверный – 0. |
| 10. Установите правильную последовательность **жизненного цикла ПО**:  1) кодирование, 2) тестирование, 3) выявление требований,  4) проектирование, 5) внедрение | | | | 3, 4, 1, 2, 5 | Верный ответ – 1 неверный – 0. |
| **11.** Установите правильную последовательность **разработки через тестирование**:  1) пишутся тесты;  2) выполняется рефакторинг;  3) пишется программный код. | | | | 1, 3, 2 | Верный ответ – 1 неверный – 0. |
| **12.** Установите правильную последовательность **жизненного цикла ошибки ПО**:  1) программист находит и исправляет дефект локально;  2) пользователь информирует программиста об ошибке;  3) программист передает исправленное ПО пользователю;  4) программист воспроизводит ошибку. | | | | 2, 4, 1, 3 | Верный ответ – 1 неверный – 0. |
| **13.** Введите ответ в поле ввода прилагательное с маленькой буквы:  При тестировании методом \_\_\_\_\_\_\_\_\_\_ ящика, разработчик теста имеет доступ к исходному коду программ. | | | | белого | Верный ответ – 1 неверный – 0. |
| **14.** Введите ответ в поле ввода (в именительном падеже с большой буквы):  \_\_\_\_\_\_\_\_\_\_ интерфейс позволяет пользователям взаимодействовать с компьютером посредством визуальных элементов, таких как значки, кнопки и окна. | | | | Графический | Верный ответ – 1 неверный – 0. |
| **15.** Введите ответ в поле ввода (в именительном падеже с маленькой буквы):  Процесс жизненного цикла ПО \_\_\_\_\_\_\_\_\_\_ определяет работы по исправлению ошибок, переносу и снятию с эксплуатации программного продукта. | | | | сопровождение | Верный ответ – 1 неверный – 0. |
| **Дисциплина «WEB -технологии» (Б1.Б.28)** | | | | | | |
| ИОПК-7.1. Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения.  **Знать:** технологии создания сайтов .  **Уметь**: выбирать технические решения и дизайн информационных систем для их работы в сети интернет в зависимости от поставленных задач и условий реализации  **Владеть:**  принципами работы в сети;  способами построение сетей в WEB;   методами реализации задач в зависимости от выбора программных комплексов на которых построены открытые информационные системы | 1. Дисциплина «WEB -технологии» направлена на получение знаний и навыков необходимых для построения в среде Internet \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Закончить фразу. | | | | WEB  приложений | Верный ответ -1 неверный – 0. |
| **2.** Укажите букву верного утверждения:  A. Сервис Angular используется для разбиения приложения на логические фрагменты кода.  Б. Модуль Angular используется для разбиения приложения на логические фрагменты кода.  В. шаблон Angular используется для разбиения приложения на логические фрагменты кода.  Г. Ничего из вышеперечисленного. | | | | Б | Верный ответ -1 неверный – 0. |
| **3.**Продолжение выражения верно под буквой \_\_ (указать): **Перехват жизненного цикла ngOnChanges вызыватся**  A. при изменении значения свойства, связанного с данными.  Б. всякий раз, когда происходит инициализация директивы / компонента после того, как Angular впервые отобразит свойства, привязанные к данным.  В. в ответ после того, как Angular проецирует внешний контент в представление компонента. | | | | А | Верный ответ -1 неверный – 0. |
| **4.** Верно, что **жизненный цикла hook – ngOnDestroy в Angular следующий**:  A. вызывается в ответ после того, как Angular проверяет содержимое, спроецированное в компонент.  Б. вызывается в ответ после того, как Angular инициализирует представления компонента и дочерние представления.  В. вызывается в ответ после того, как Angular проверяет представления компонента и дочерние представления.  Г. вызывается непосредственно перед тем, как Angular уничтожит директиву / компонент. | | | | Г | Верный ответ -1 неверный – 0. |
| **5.** Выберите два утверждения, которые **верны для директивы let в JavaScript:**  А. Областью видимости переменных, объявленных директивой let, является блок, в котором они объявлены, исключая все его подблоки.  Б. Областью видимости переменных, объявленных директивой let, является блок, в котором они объявлены, и все его подблоки.  В. Директива let объявляет переменную с блочной областью видимости и с невозможностью инициализировать её значением при объявлении.  Д. Директива let объявляет переменную с блочной областью видимости с возможностью инициализировать её значением при объявлении. | | | | Б  Д | Верный ответ -1 неверный – 0. |
| **6.** Допишите недостающее с большой буквы:  Для взаимодействия с сервером и отправки запросов по протоколу http применяется класс \_\_\_\_\_\_\_\_\_\_\_\_\_. | | | | HttpClient. | Верный ответ -1 неверный – 0. |
| **7.** (**dir)** определяет направление текста: слева направо (ltr) или справа налево \_\_\_\_\_\_\_\_. | | | | **(rtl)**. | Верный ответ –1 неверный – 0. |
| **8.** Допишите недостающий компонент.  Директива – это настраиваемый элемент HTML, Директивы делятся на атрибутивные и \_\_\_\_\_\_\_\_\_\_\_\_\_ компоненты. | | | | структурные | Верный ответ –1 неверный – 0. |
| **9.** Укажите порядок действий для создания нового проекта Angular в Visual Studio Code:   1. Создайте новый проект с помощью команды: `ng new angular-example`. 2. Запустите проект: выберите опцию «Terminal» в меню «View» и введите команду `npm start`. 3. Откройте созданный проект в VS Code: выберите вкладку «Explorer» на боковой панели или нажмите Ctrl + Shift + E. 4. Установите Angular CLI с помощью команды: `npm install -g @angular/cli`. | | | | 4, 1, 3, 2 | Верный ответ –1 неверный – 0. |
| 10. При написании HTML кода Web-страницы названия (имена, идентификаторы) тегов заключаются в скобки.  Укажите словом вид скобок | | | | угловые | Верный ответ –1 неверный – 0. |
| **11.** Допишите недостающие части определений с маленькой буквы через дефис:  Сервер, который играет роль посредника между рабочей станцией пользователя и Интернетом, и позволяет повысить безопасность, административный контроль и использовать функции кэширования, это … | | | | прокси-сервер | Верный ответ -1 неверный – 0. |
| **12.** Порядок действий для настройки сервера Angular в Visual Studio Code состоит из команд.  Укажите цифрой первую и последнюю команды   1. Перейдите в раздел «Файл» > «Открыть папку». 2. Откройте Visual Studio Code. 3. Перейдите в папку <username>/Desktop/Angular-Projects/my-app/ и выберите «Выбрать папку». 4. Откройте встроенный терминал. Для этого перейдите в раздел «Терминал» > «Новый терминал» или «Просмотр» > «Терминал». 5. Введите команду `ng serve` и нажмите Enter. | | | | 2, 5 | Верный ответ -1 неверный – 0. |
| **13.** Укажите порядок действий для создания нового проекта Angular в WebStorm:   1. Укажите местоположение проекта. 2. Нажмите «Создать новый проект» на экране приветствия или выберите «Файл» | «Новый» | «Проект» в главном меню. 3. В открывшемся диалоговом окне выберите «Пустой проект» на левой панели. | | | | 2, 3, 1 | Верный ответ –1 неверный – 0. |
| **14.** Сопоставьте определения и их значения:   |  |  |  |  | | --- | --- | --- | --- | | А | Директивы | 1 | используется для детализации того, как каждая часть приложения подходит друг другу. | | Б | Привязка данных | 2 | работают для того, чтобы связать определенное поведение с различными элементами в DOM | | В | NgModule | 3 | обеспечивает связь между DOM, который вы разрабатываете, и компонентами, и оно состоит из четырех основных форм. | | Г | Область видимости | 4 | это просто объект, который представляет всю модель приложения. | | | | | |  |  | | --- | --- | | А | 2 | | Б | 3 | | В | 1 | | Г | 4 | | Верный ответ –1 неверный – 0. |
| **15.** Сопоставьте компоненты из Angular 2 и их определения:   |  |  |  |  | | --- | --- | --- | --- | | А | Templates | 1 | используется для разбиения приложения на логические части кода. Каждый фрагмент кода или модуль предназначен для выполнения одной задачи. | | Б | Component | 2 | можно использовать для объединения модулей | | В | Service | 3 | используется для определения представлений приложения Angular | | Г | Modules | 4 | используется для создания компонентов, которые могут использоваться во всем приложении. | | | | | |  |  | | --- | --- | | А | 3 | | Б | 1 | | В | 4 | | Г | 2 | | Верный ответ –1 неверный – 0. |
| **Дисциплина «Проектирование системного программного обеспечения» (Б1.Б.29)** | | | | | | |
| ИОПК-7.1. Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения  **Знать:**  − устройство и структуру платформ виртуализации кода, принципы их работы (ИОПК -7.1.)  − базовый синтаксис и приемы языка программирования С# (ИОПК-7.1.)  **Уметь:**  − осуществлять настройку платформ под конкретную конфигурацию оборудования  (ИОПК -7.1.)  − разрабатывать программное обеспечение под платформу .NET  (ИОПК-7.1.)  **Владеть:**  − методами и средствами администрирования и настраивания системного программного обеспечения (ИОПК-7.1.)  − средствами разработки программного обеспечения под платформу .NET  (ИОПК -7.1.) | **1. Что такое системное программное обеспечение?**  1. Программы для работы в Интернете  2. Программное обеспечение, управляющее аппаратными ресурсами и обеспечивающее работу приложений  3. Игровые приложения | | | | 2 | Верный ответ 1, неверный – 0. |
| **2. Какова основная цель проектирования системного ПО?**  1. Создание интерфейсов  2. Разработка надежного, эффективного и безопасного программного обеспечения  3. Увеличение объема памяти | | | | 2 | Верный ответ 1, неверный – 0. |
| **3. Какие этапы включает проектирование системного ПО?**  1. Только тестирование  2. Анализ требований, проектирование архитектуры, реализация, тестирование, поддержка  3. Одна итерация разработки | | | | 2 | Верный ответ 1, неверный – 0. |
| **4. Что такое модульность в проектировании?**  1. Разделение системы на подсистемы  2. Разбиение программы на независимые, но взаимодействующие компоненты  3. Увеличение объема программного кода | | | | 2 | Верный ответ 1, неверный – 0. |
| **5. Почему важно учитывать масштабируемость при проектировании системного ПО?**  1. Чтобы система могла поддерживать увеличение нагрузки и роста данных  2. Для уменьшения затрат  3. Для создания интерфейсов | | | | 1 | Верный ответ 1, неверный – 0. |
| **6. Что такое архитектура программного обеспечения?**  1. Код программы  2. Структура системы, включающая компоненты и их взаимодействие  3. Алгоритм сортировки данных | | | | 2 | Верный ответ 1, неверный – 0. |
| **7. Чем отличается монолитная архитектура от микросервисной?**  1. Микросервисы всегда быстрее  2. Монолитная система — единое приложение, микросервисы — набор независимых модулей  3. У них одинаковая структура | | | | 2 | Верный ответ 1, неверный – 0. |
| **8. Что такое распределенные системы?**  1. Системы без взаимодействия компонентов  2. Системы, работающие на нескольких узлах, взаимодействующих между собой  3. Упрощенные приложения | | | | 2 | Верный ответ 1, неверный – 0. |
| **9. Что такое интерфейс прикладного программирования (API)?**  1. Набор правил и инструментов для взаимодействия между программными компонентами  2. Алгоритм сжатия данных  3. Среда разработки программ | | | | 1 | Верный ответ 1, неверный – 0. |
| **10. Что такое проектирование на основе шаблонов?**  1. Использование готовых решений для типовых задач проектирования  2. Автоматизация тестирования  3. Создание пользовательских интерфейсов | | | | 1 | Верный ответ 1, неверный – 0. |
| **11. Почему важна производительность системного ПО?**  1. Для увеличения объема данных  2. Для эффективного использования ресурсов и обеспечения быстродействия системы  3. Для создания интерфейсов | | | | 2 | Верный ответ 1, неверный – 0. |
| **12. Как измеряется производительность ПО?**  1. Количеством пользователей  2. Временем отклика, использованием ресурсов, пропускной способностью  3. Размером кода | | | | 2 | Верный ответ 1, неверный – 0. |
| **13. Что такое отказоустойчивость?**  1. Автоматическая замена модулей  2. Способность системы продолжать работать при сбоях  3. Снижение производительности | | | | 2 | Верный ответ 1, неверный – 0. |
| **14. Какие методы используются для повышения надежности ПО?**  1. Только тестирование  2. Резервирование, обработка ошибок, контроль целостности данных  3. Исключение избыточных данных | | | | 2 | Верный ответ 1, неверный – 0. |
| **15. Какую роль играет тестирование в проектировании системного ПО?**  1. Увеличивает объем кода  2. Обнаруживает ошибки и проверяет соответствие требованиям  3. Исключает повторное использование кода | | | | 2 | Верный ответ 1, неверный – 0. |
| **16. Почему важно учитывать безопасность в системном ПО?**  1. Для защиты системы и данных от угроз и атак  2. Для ускорения обработки данных  3. Для улучшения интерфейсов | | | | 1 | Верный ответ 1, неверный – 0. |
| **17. Что такое принцип наименьших привилегий?**  1. Ограничение прав доступа только необходимым минимумом  2. Предоставление полного доступа  3. Автоматическое обновление данных | | | | 1 | Верный ответ 1, неверный – 0. |
| **18. Какое значение имеет документирование при проектировании ПО?**  1. Увеличивает время разработки  2. Обеспечивает понимание системы разработчиками и пользователями  3. Ускоряет обработку данных | | | | 2 | Верный ответ 1, неверный – 0. |
| **19. Как поддержка влияет на жизненный цикл системного ПО?**  1. Увеличивает объем работы  2. Обеспечивает устранение ошибок, обновление и адаптацию к новым условиям  3. Снижает надежность системы | | | | 2 | Верный ответ 1, неверный – 0. |
| **20. Что такое рефакторинг?**  1. Полная переработка программы  2. Улучшение структуры кода без изменения его функциональности  3. Оптимизация ресурсов | | | | 2 | Верный ответ 1, неверный – 0. |
| **Дисциплина «Скриптовые языки программирования» (Б1.Б.30)** | | | | | | |
| ИОПК-7.1. Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения  **Знать:**  − отличительные особенности и области применения скриптовых языков программирования (ИОПК -7.1.)  − базовые особенности реализации принципов структурного и объектно-ориентированного программирования на таких языках (ИОПК-7.1.)  **Уметь:**  − выбирать язык программирования как инструмент решения конкретной практической задачи, планировать и реализовывать решение задачи на скриптовом языке программирования (ИОПК-7.1.)  **Владеть:**  − методами по разработке программ на языке Python  (ИОПК -7.1.) | **1. Что такое скриптовый язык программирования?**  1. Язык, предназначенный для автоматизации задач и управления другими программами  2. Язык для создания игр  3. Язык для разработки операционных систем | | | | 2 | Верный ответ 1, неверный – 0. |
| **2. Какая особенность скриптовых языков отличает их от компилируемых?**  1. Код всегда пишется на уровне машинных инструкций  2. Код выполняется интерпретатором без предварительной компиляции  3. Они работают только на серверах | | | | 2 | Верный ответ 1, неверный – 0. |
| **3. Какова основная цель использования скриптовых языков?**  1. Создание аппаратного обеспечения  2. Автоматизация, прототипирование и интеграция  3. Построение сетевых протоколов | | | | 2 | Верный ответ 1, неверный – 0. |
| **4. Какие из этих языков являются скриптовыми?**  1. Python, JavaScript, Ruby  2. C, C++, Java  3. Pascal, Assembly, Fortran | | | | 2 | Верный ответ 1, неверный – 0. |
| **5. Чем отличается интерпретатор от компилятора?**  1. Интерпретатор сразу исполняет код, а компилятор сначала преобразует его в исполняемый файл  2. Интерпретатор выполняет программы построчно, компилятор — полностью  3. Интерпретатор быстрее компилятора | | | | 2 | Верный ответ 1, неверный – 0. |
| **6. Для чего часто используют JavaScript?**  1. Разработка баз данных  2. Создание интерактивных веб-приложений  3. Управление серверами | | | | 2 | Верный ответ 1, неверный – 0. |
| **7. Что такое динамическая типизация?**  1. Возможность языка автоматически определять тип переменной во время выполнения программы  2. Жёсткое определение типов на этапе компиляции  3. Метод сжатия данных | | | | 2 | Верный ответ 1, неверный – 0. |
| **8. В каком случае используются регулярные выражения?**  1. Для работы со строками: поиска, проверки шаблонов и замены  2. Для создания баз данных  3. Для работы с сетевыми протоколами | | | | 2 | Верный ответ 1, неверный – 0. |
| **9. Какие преимущества даёт использование библиотек в скриптовых языках?**  1. Ускоряет работу интерпретатора  2. Расширяет функциональность языка, упрощая выполнение сложных задач  3. Уменьшает объём кода | | | | 2 | Верный ответ 1, неверный – 0. |
| **10. Что такое серверные скрипты?**  1. Скрипты для управления браузером  2. Программы, выполняемые на сервере для обработки запросов клиентов  3. Скрипты для сжатия данных | | | | 2 | Верный ответ 1, неверный – 0. |
| **11. Какой язык является стандартом для работы с веб-браузерами?**  1. PHP  2. JavaScript  3. Python | | | | 2 | Верный ответ 1, неверный – 0. |
| **12. Для чего чаще всего используют язык PHP?**  1. Управление графикой  2. Создание серверных веб-приложений  3. Разработка мобильных приложений | | | | 2 | Верный ответ 1, неверный – 0. |
| 13. **Что делает функция eval()?**  1. Компилирует программу  2. Выполняет строку кода, переданную в неё, как программу  3. Оптимизирует код | | | | 2 | Верный ответ 1, неверный – 0. |
| **14. Какая основная задача языка Bash?**  1. Разработка приложений  2. Автоматизация работы в командной строке Unix/Linux  3. Обработка изображений | | | | 2 | Верный ответ 1, неверный – 0. |
| **15. В чём преимущество Python перед многими другими скриптовыми языками?**  1. Он быстрее всех языков  2. Он обладает широкой библиотекой модулей и простым синтаксисом  3. Подходит только для веб-разработки | | | | 2 | Верный ответ 1, неверный – 0. |
| **16. В чём преимущество интерпретации скриптов?**  1. Отсутствие необходимости компиляции позволяет быстрее начать выполнение  2. Увеличение скорости выполнения  3. Полное исключение ошибок | | | | 2 | Верный ответ 1, неверный – 0. |
| **17. В чём ограничение скриптовых языков?**  1. Они медленнее компилируемых языков из-за интерпретации  2. Невозможность работы с сетевыми протоколами  3. Они не поддерживают циклы | | | | 2 | Верный ответ 1, неверный – 0. |
| **18. Почему скриптовые языки популярны для прототипирования?**  1. Они создают визуальные прототипы  2. Они позволяют быстро разрабатывать и тестировать программы  3. Они автоматически оптимизируют код | | | | 2 | Верный ответ 1, неверный – 0. |
| **19. Какой недостаток может быть у функции eval()?**  1. Она медленно работает  2. Она может быть небезопасной, если принимает непроверенные данные  3. Она не поддерживает циклы | | | | 2 | Верный ответ 1, неверный – 0. |
| **20. Почему скриптовые языки популярны в веб-разработке?**  1. Они легко интегрируются с браузерами и серверами  2. Они работают быстрее компилируемых языков  3. Они имеют встроенные графические инструменты | | | | 2 | Верный ответ 1, неверный – 0. |
| **Дисциплина «Алгоритмы и структуры данных» (Б1.Б.42)** | | | | | |  |
| **ИОПК -7.1.**  Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения.  **Знать:**  основные структуры данных, рекурсивные алгоритмы ;  методы работы с рекурсивными алгоритмами;  алгоритмические методы работы с разреженными матрицами;  - алгоритмами быстрого поиска и сортировки.  Уметь:  разрабатывать специальные структуры данных ;  проводить отладку и тестирование программных модулей;  строить современные автоматизированные системы обработки информации.  Владеть:  современными инструментами разработки и отладки программного обеспечения. |
| **1.** Список, каждый элемент которого содержит указатели на следующий и предыдущий элементы, называют  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | двусвязный | Верный ответ 1, неверный – 0. |
| 1. Ситуация, когда для двух различных ключей xi ≠ xj значения хэш - функций равны h(xi)=h(xj) называется   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | коллизия | Верный ответ 1, неверный – 0. |
| **3.** Специальный случай рекурсии, при котором рекурсивный вызов функцией самой себя является её последней операцией, носит название \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | хвостовая | Верный ответ 1, неверный – 0. |
| **4.** Процедура, при которой выделяются все элементы, от которых зависит решение, и, в частности размерность рекурсивной задачи, носит название \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | параметризация | Верный ответ –1 неверный – 0. |
| **5.** Установите соответствие между названием и определением информационного объекта:   |  |  |  |  | | --- | --- | --- | --- | | А | Деревья, каждая вершина которых имеет не более двух сыновей и сбалансированные по высоте | 1 | AVL - деревья | | Б | Деревья, каждая вершина которых имеет не более двух сыновей, в которых каждая вершина имеет две координаты | 2 | Бинарные деревья | | В | Деревья, каждая вершина которых имеет не более двух сыновей | 3 | Декартовые деревья | | | | | |  |  | | --- | --- | | А | 1 | | Б | 3 | | В | 2 | | Верный ответ 1, неверный – 0. |
| **6.** Установите соответствие между названием и определением функции:   |  |  |  |  | | --- | --- | --- | --- | | А | Не рекурсивная функция | 1 | В функции есть вызов той же самой функции | | Б | Прямая рекурсия функция | 2 | В функции нет вызова той же самой функции | | В | Косвенная рекурсия функция | 3 | В теле функции нет вызова той же самой функции, но она обращается к другой функции, которая обращается к первой | | | | | |  |  | | --- | --- | | А | 2 | | Б | 1 | | В | 3 | | Верный ответ 1, неверный – 0. |
| **7.**Установите соответствие между названием и определением типа хэш-функции:   |  |  |  |  | | --- | --- | --- | --- | | А | Хэш-функция деления | 1 | k = k\*k; k >>= n1; H = key % n2 ; | | Б | Хэш-функция умножения | 2 | H = (k) mod m, где k - ключ, m - размер хеш-таблицы. | | В | Хэш-функция «середины квадрата» | 3 | H ={ m\* ((C • K) mod 1) }  Оператор { } возвращает наибольшее целое, которое меньше аргумента. | | | | | |  |  | | --- | --- | | А | 2 | | Б | 3 | | В | 1 | | Верный ответ 1, неверный – 0. |
| **8.** Установите соответствие между названием и определением информационного объекта:   |  |  |  |  | | --- | --- | --- | --- | | А | Вершина (узел) | 1 | Объект дерева, содержащий данные и соединенный с другими такими же объектами с помощью ветвей | | Б | Брат (сестра) | 2 | Терминальная вершина | | В | Корень | 3 | Самая верхняя вершина дерева, не имеющая родительских вершин | | Г | Лист | 4 | Две вершины дерева, имеющие общего родителя | | | | | |  |  | | --- | --- | | А | 1 | | Б | 4 | | В | 3 | | Г | 2 | | Верный ответ 1, неверный – 0. |
| **9.** Установите правильную последовательность алгоритмической производительности алгоритма по возрастанию:  1) сложность O(1),  2) сложность O(Log (N)),  3) сложность O(N). | | | | 1, 2, 3 | Верный ответ 1, неверный – 0. |
| **10.** Установите правильную последовательность алгоритма быстрой сортировки:  1) выберем некоторый ключ (главный элемент);  2) поставим главный элемент между двумя этими группами;  3) все ключи разделим на две группы: ключи, меньшие главный элемент и большие главный элемент;  4) применим к каждой из групп тот же самый алгоритм. | | | | 1, 3, 2, 4 | Верный ответ 1, неверный – 0. |
| **11.** Установите правильную последовательность этапов решения игровой программы «Игры двух игроков»:  1) построение оценочной функции,  2) построение функции Альфа-бетта,  3) построение функции МиниМакс. | | | | 1, 3, 2 | Верный ответ 1, неверный – 0. |
| **12.** Установите правильную последовательность прохождения бинарного дерева в глубину в прямом порядке:  1) посещение корня и его обработка,  2) рекурсивное посещение левого поддерева,  3) рекурсивное посещение правого поддерева. | | | | 1, 2, 3 | Верный ответ 1, неверный – 0. |
| **13.** Введите ответ в поле ввода (в именительном падеже с большой буквы):  \_\_\_\_\_\_\_\_\_\_ – это набор команд, необходимых для решения той или иной задачи. | | | | Алгоритм | Верный ответ 1, неверный – 0. |
| **14.** Введите ответ в поле ввода (в именительном падеже с большой буквы):  \_\_\_\_\_\_\_\_\_\_ – это список, в котором включение элементов выполняется с одного конца, а исключения элементов осуществляются с другого конца. | | | | Очередь | Верный ответ 1, неверный – 0. |
| **15.** Введите ответ в поле ввода в именительном падеже с большой буквы (Какая матрица?)  \_\_\_\_\_\_\_\_\_\_ матрица – это матрица, в которой количество ненулевых элементов много меньше общего числа элементов. | | | | Разреженная | Верный ответ 1, неверный – 0. |
|  | **Дисциплина «Объектно-ориентированное программирование» (Б1.Б.47)** | | | | | |
| ИОПК-7.1.  Применяет современные языки программирования, технологии и инструмен-ты для создания програм-много обеспечения.  **Знать:**  основные концепции объектно-ориентированного программирования (инкапсуляция, наследование, обработка исключений) (язык Java);  паттерны проектирова-ния (язык Java).  **Уметь:**  проводить отладку и тестирование програм-мных модулей на языке Java;  разрабатывать программное обеспечение для автоматизированные системы обработки информации на языке Java.  **Владеть:**  методами обработки исключительных ситуаций на языке Java | **1.**Объектно-ориентированное программированиестроится на объектах и классах.  Класс это- *дать определение* | | | | тип данных, созданных пользователем | Верный ответ 1, неверный – 0. |
| **2.** Объект — это экземпляр класса, или его копия, которая находится в \_\_\_\_\_\_\_\_\_. | | | | памяти компьютера | Верный ответ 1, неверный – 0. |
| **3**.Сколько объектов может быть у имеющегося Класса? | | | | Много или не  одного | Верный ответ 1, неверный – 0. |
| 1. Вставьте нужное слово   Создание объекта происходит путем вызова \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ класса. | | | | конструктора | Верный ответ 1, неверный – 0. |
| 1. Определите соответствия  |  |  |  |  | | --- | --- | --- | --- | | 1 | метод | А | способы реакции объекта на событие | | 2 | событие | Б | характеристики объекта | | 3 | свойство | В | то, что происходит в реальном мире и определяет поведение объекта | | | | | А-1  Б-3  В-2 | Верный ответ 1, неверный – 0. |
|  | | | |  |  |
| 1. Средой взаимодействия объектов являются \_\_\_\_\_\_\_\_\_\_\_\_\_ (вписать слово во множественном числе) | | | | сообщения | Верный ответ 1, неверный – 0. |
| **7.** Установите соответствие видов тестирования их назначению:   |  |  |  |  | | --- | --- | --- | --- | | А | тестирование методом «белого ящика» | 1 | тестирование применимости программной системы с использованием региональных настроек | | Б | тестирование совместимости | 2 | проверка корректной работы программы в определенном окружении | | В | тестирование локализации | 3 | разработка тестов с учетом исходного кода | | | | | |  |  | | --- | --- | | А | 3 | | Б | 2 | | В | 1 | | Верный ответ 1, неверный – 0. |
| 1. Установите соответствие диаграмм языка UML их назначению:  |  |  |  |  | | --- | --- | --- | --- | | А | Диаграмма классов | 1 | показывают множество классов, интерфейсов и отношений между ними | | Б | Диаграмма объектов | 2 | показывает последовательность сообщений и структуру взаимодействия объектов | | В | Диаграмма кооперации | 3 | показывает множество объектов и отношений между ними в некоторый момент времени | | | | | |  |  | | --- | --- | | А | 1 | | Б | 3 | | В | 2 | | Верный ответ 1, неверный – 0. |
| **8.** В какой последовательности осуществляется разработка класса?   1. Определяется внешний интерфейс класса. 2. Выполняется реализация методов класса. 3. Определяется ответственность класса. | | | | 3, 1, 2 | Верный ответ 1, неверный – 0. |
| **9**. Установите правильную последовательность проектирования программной системы:  1) разделение подсистем на классы; 2) разделение системы на подсистемы и компоненты;  3) разделение классов на данные и методы; 4) проектирование методов. | | | | 2, 1, 3, 4 | Верный ответ 1, неверный – 0. |
| **10.** Установите правильную последовательность предпочтительности способов применения наследования:  1) для многократного использования;  2) для отличий;  3) для замены типов. | | | | 3, 2, 1 | Верный ответ 1, неверный – 0. |
| **11.** Установите правильную последовательность тестирования программной системы:  1) бета-тестирование (тестирование пользователей);  2) функциональное тестирование; 3) модульное тестирование;  4) тестирование совместимости. | | | | 3, 2, 4, 1 | Верный ответ 1, неверный – 0. |
| **12.** Впишите ответ с маленькой буквы:  Инкапсуляция требует, чтобы большинство методов и членов данных класса определялась с областью видимости  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | private | Верный ответ 1, неверный – 0. |
| **13.** Впишите ответ с маленькой буквы: Для обеспечения доступа к методам и членам класса только для потомков данного класса их необходимо объявить с ключевым словом \_\_\_\_\_\_\_\_\_ | | | | protected | Верный ответ 1, неверный – 0. |
| **14.** Впишите ответ с маленькой буквы в именительном падеже:  Структурное отношение между классами в UML, показывающее, что объекты одного типа неким образом связаны с объектами другого типа, называется \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | ассоциация | Верный ответ 1, неверный – 0. |
| **15.** Впишите ответ с маленькой буквы в именительном падеже:  Отношение между классами в программе, показывающее, что объекты одного типа являются также объектами другого типа, называется \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | наследование | Верный ответ 1, неверный – 0. |
| **Дисциплина «Программная инженерия» (Б1.Б.48 )** | | | | | | |
| **ИОПК-7.1.**  Применяет современные языки программирования, технологии и инструменты для создания программного обеспечения.  **Знать:**  − современные технологии управления программными проектами. (ИОПК-7.1)  − гибкие технологии для управления разработкой программного обеспечения информационных систем (ИОПК -7.1)  **Уметь:**  − разрабатывать программные средства с помощью современных технологий управления программными проектами. (ИОПК -7.1) − применять гибкие технологии для управления разработкой программного обеспечения информационных систем (ИОПК -7.1)  **Владеть:**  − навыками разработки и модернизации программного обеспечения информационных систем на основе гибких технологий. (ИОПК -7.1) | 1. Что такое программная инженерия?  1. Наука о проектировании, разработке, тестировании и поддержке программных систем  2. Процесс программирования на определенном языке  3. Ремонт и модернизация программных продуктов | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 2. Какую задачу решает программная инженерия?  1. Обеспечение качества программного обеспечения на всех этапах его жизненного цикла  2. Ремонт программных ошибок  3. Разработка новых языков программирования | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 3. Каковы основные этапы разработки программного обеспечения?  1. Анализ требований, проектирование, реализация, тестирование, поддержка  2. Разработка интерфейса, сборка и развертывание  3. Описание функционала и создание документации | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 4. Что такое жизненный цикл разработки программного обеспечения?  1. Последовательность этапов от первоначальной идеи до поддержки и завершения разработки программы  2. Процесс оптимизации работы программы  3. Тестирование программы перед выпуском | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 5. Какие методологии разработки программного обеспечения существуют?  1. Каскадная модель, Agile, DevOps, Scrum  2. Модели проектирования интерфейсов  3. Статические методы оптимизации | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 6. Что такое методология Agile?  1. Методология разработки, основанная на гибкости, частых изменениях и взаимодействии с заказчиком  2. Методология для написания документации  3. Статическая модель разработки | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 7. Какое место занимает тестирование в программной инженерии?  1. Тестирование является важной частью разработки, направленной на обнаружение ошибок и обеспечение качества ПО  2. Тестирование проводится только на этапе завершения разработки  3. Тестирование важнее всех этапов разработки | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 8. Что такое системный анализ в контексте программной инженерии?  1. Изучение и анализ требований заказчика для разработки эффективного программного решения  2. Процесс создания интерфейсов и графиков  3. Метод разработки новой операционной системы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 9. Какова цель проектирования в программной инженерии?  1. Создание архитектуры программного продукта, которая будет эффективной, надежной и удобной для пользователей  2. Написание исходного кода  3. Формирование требований к системе | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 10. Что такое моделирование в программной инженерии?  1. Представление системы в виде абстракции для анализа, проектирования и проверки решений  2. Процесс оптимизации кода  3. Создание интерфейсов и макетов | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 11. Что такое "риск-менеджмент" в программной инженерии?  1. Процесс выявления, анализа и управления рисками, которые могут повлиять на успешную разработку программного обеспечения  2. Ремонт ошибок в коде  3. Поддержка устаревших технологий | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 12. Что такое "технический долг" в программной инженерии?  1. Неоправданные или отложенные решения, которые могут замедлить дальнейшее развитие системы  2. Неработающее приложение  3. Ошибки, которые исправляются в будущем | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 13. Каковы преимущества применения метода "Scrum" в разработке программного обеспечения?  1. Позволяет создавать программное обеспечение через небольшие итерации, активно вовлекая команду и заказчика  2. Упрощает написание документации  3. Снижает требования к ресурсам | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 14. Что такое "повторное использование компонентов" в программной инженерии?  1. Использование уже существующих и проверенных компонентов или кода для ускорения разработки  2. Разработка новых языков программирования  3. Ремонт устаревших компонентов | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 15. Какие особенности существуют при разработке программного обеспечения для крупных распределенных систем?  1. Необходимость учета сложных сетевых взаимодействий, безопасности и синхронизации данных  2. Упрощение структуры системы  3. Отсутствие необходимости в тестировании | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 16. Что такое "интеграционное тестирование"?  1. Процесс проверки взаимодействия компонентов системы и их корректности при объединении  2. Тестирование пользовательского интерфейса  3. Периодическое тестирование скорости работы системы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 17. Какую роль играют стандарты в программной инженерии?  1. Обеспечивают единые правила и требования для разработки, тестирования и внедрения программных продуктов  2. Помогают улучшить скорость работы программ  3. Стандарты не имеют значения в программной инженерии | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 18. Что такое "код-ревью"?  1. Процесс коллективной проверки исходного кода другими разработчиками для поиска ошибок и улучшения качества  2. Поддержка старых версий программ  3. Проверка сетевой безопасности | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 19. Каковы основные принципы проектирования программного обеспечения?  1. Модульность, абстракция, повторное использование, независимость компонентов  2. Только ориентированность на производительность  3. Ремонт и обновление кода | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 20. Что такое "программное обеспечение с открытым исходным кодом"?  1. Программное обеспечение, исходный код которого доступен для использования, изменения и распространения  2. Программное обеспечение с ограниченным доступом  3. Программное обеспечение, поддерживающее только один язык программирования | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| **Дисциплина «Тестирование программного обеспечения» (Б1.Б.49)** | | | | | | |
| **ИОПК -7.2.**  Выполняет выбор инструментов и технологий для обеспечения качества программного обеспечения.  **Знать:**  − принципы тестирования (ИОПК -7.2)  − типы, виды и уровни тестирования (ИОПК -7.2 )  − характеристики качества (ИОПК -7.2)  **Уметь:**  − проводить компонентное, интеграционное, системное и приемочное тестирование (ИОПК -7.2)  **Владеть:**  − навыками ручного и автоматизированного тестирования. (ИОПК -7.2) | 1. Что такое тестирование программного обеспечения?  1. Процесс проверки и оценки качества программного продукта путем выявления ошибок и дефектов  2. Процесс написания исходного кода  3. Процесс разработки новых функций программы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 2. Какую цель преследует тестирование программного обеспечения?  1. Обнаружение ошибок и дефектов, обеспечение качества и надежности программы  2. Улучшение внешнего вида программы  3. Повышение производительности компьютера | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 3. Какова основная цель юнит-тестирования?  1. Проверка отдельных компонентов программы на их корректную работу  2. Тестирование интерфейса пользователя  3. Тестирование безопасности системы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 4. Что такое интеграционное тестирование?  1. Процесс проверки взаимодействия между модулями или компонентами системы  2. Тестирование производительности  3. Тестирование всех компонентов в реальной рабочей среде | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 5. Что такое системное тестирование?  1. Проверка всей системы как целого, чтобы убедиться в ее соответствии функциональным требованиям  2. Тестирование кода на уровне отдельных функций  3. Тестирование пользовательского интерфейса | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 6. Какую роль играет тестирование на этапе разработки программного обеспечения?  1. Оно помогает найти и исправить ошибки на ранних стадиях разработки, предотвращая их распространение в готовом продукте  2. Оно ускоряет процесс создания кода  3. Оно помогает улучшить графику приложения | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 7. Что такое функциональное тестирование?  1. Тестирование, которое фокусируется на проверке, соответствует ли программное обеспечение заявленным функциональным требованиям  2. Тестирование пользовательского интерфейса  3. Тестирование производительности и масштабируемости | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 8. Какую задачу решает тестирование производительности программного обеспечения?  1. Проверка, как программа ведет себя под высокой нагрузкой и насколько она эффективно использует ресурсы  2. Проверка интерфейса пользователя  3. Тестирование локализации программы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 9. Что такое тестирование безопасности?  1. Процесс проверки уязвимостей программы, чтобы убедиться, что она защищена от атак и утечек данных  2. Проверка производительности программы  3. Тестирование локализации программного обеспечения | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 10. Что такое альфа-тестирование?  1. Тестирование, проводимое разработчиками на ранней стадии разработки перед передачей в тестовую команду  2. Тестирование программы на реальных пользователях  3. Тестирование на внешних условиях | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 11. Что такое бета-тестирование?  1. Тестирование, проводимое внешними пользователями, с целью получения обратной связи и выявления ошибок в реальных условиях  2. Тестирование в процессе разработки  3. Проводится только внутри компании-разработчика | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 12. Какую роль играет автоматизированное тестирование в процессе разработки?  1. Автоматизированное тестирование ускоряет проверку повторяющихся сценариев и помогает обеспечить стабильность продукта  2. Оно заменяет необходимость в ручном тестировании  3. Оно используется только для функциональных тестов | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 13. Что такое тестирование пользовательского интерфейса (UI)?  1. Проверка удобства и функциональности интерфейса, а также корректности отображения элементов на экране  2. Проверка работы базы данных  3. Тестирование производительности системы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 14. Что такое регрессионное тестирование?  1. Тестирование, которое проверяет, не нарушены ли ранее работавшие функции программы после внесения изменений  2. Проверка новых функций  3. Тестирование безопасности системы | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 15. Что такое тестирование совместимости?  1. Проверка, как программа работает на различных операционных системах, устройствах или в разных браузерах  2. Тестирование только производительности  3. Проверка производительности на одном устройстве | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 16. Какие инструменты обычно используются для автоматизированного тестирования?  1. Selenium, JUnit, TestNG, Appium  2. Visual Studio  3. Программы для мониторинга безопасности | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 17. Какой из следующих типов тестирования фокусируется на пользовательском опыте?  1. Юзабилити-тестирование  2. Тестирование безопасности  3. Тестирование функциональности | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 18. Что такое "черный ящик" в контексте тестирования?  1. Метод тестирования, при котором тестировщик не знает внутренней реализации системы и проверяет ее только с точки зрения функциональности  2. Тестирование, при котором проверяется только внутренняя структура кода  3. Метод тестирования с подробным анализом исходного кода | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 19. Что такое "белый ящик" в контексте тестирования?  1. Метод тестирования, при котором тестировщик имеет доступ к исходному коду и проверяет его на наличие ошибок  2. Тестирование только пользовательского интерфейса  3. Тестирование на реальных устройствах | | | | 1 | Верный ответ – 1 балл, неверный – 0. |
| 20. Что такое тестирование на отказоустойчивость?  1. Проверка, как программа продолжает работать в случае отказа отдельных компонентов или проблем с системой  2. Тестирование только производительности  3. Проверка на безопасность приложения | | | | 1 | Верный ответ – 1 балл, неверный – 0. |