course-logo.png

***Из предложенных вариантов ответов выберите один.***

|  |  |  |
| --- | --- | --- |
| **1.** | **Что не может быть причиной отказа программы?** |  |
| 1. Дефект в коде |  |
| 1. Неправильное использование программы |  |
| 1. Условия окружающей среды |  |
| 1. Недочеты в программно-аппаратных средствах |  |
| **2.** | **Всегда ли дефект в программном коде приводит к отказу?** |  |
| 1. Да |  |
| 1. Нет |  |
| **3.** | **Что из ниже перечисленного неправда?** |  |
| 1. Тестирование не может доказать, что дефектов нет |  |
| 1. Исчерпывающее тестирование недостижимо |  |
| 1. Тестирование должно быть начато только после того, как стадия разработки окончена |  |
| 1. Если одни и те же тесты будут прогоняться много раз, в конечном счете, этот набор тестовых сценариев больше не будет находить новых дефектов |  |
| **4.** | **Какие из методологий разработки ПО не относятся к гибким?** |  |
| 1. Scrum |  |
| 1. XP |  |
| 1. RUP |  |
| 1. Kanban |  |
| **5.** | **Что из нижеперечисленного не относится к характеристикам качественного тестирования в любой модели жизненного цикла ПО?** |  |
| 1. Каждому процессу разработки соответствует свой процесс тестирования |  |
| 1. Каждый уровень тестирования имеет свои цели тестирования |  |
| 1. Анализ и дизайн тестов для какого-либо уровня тестирования должны начинаться одновременно с соответствующей деятельностью разработчиков |  |
| 1. Тестировщики не должны быть вовлечены в процесс просмотра и рецензирования документов |  |
| **6.** | **Что из нижеперечисленного не является разновидностью приемочного тестирования?** |  |
| 1. Alpha |  |
| 1. Field |  |
| 1. System integration |  |
| 1. Beta |  |
| **7.** | **Что из нижеперечисленного чаще является правдой?** |  |
| 1. Компонентное тестирование – важная часть приемочного тестирования |  |
| 1. Цель компонентного тестирования – выявить проблемы в интеграции между компонентами ПО и аппаратным обеспечением |  |
| 1. Инспекции кода часто используются в компонентном тестировании |  |
| **8.** | **Что не относится к нефункциональным характеристикам ПО?** |  |
| 1. Безопасность |  |
| 1. Производительность |  |
| 1. Надежность |  |
| 1. Удобство использования |  |
| **9.** | Для кода, написанного ниже, определите минимальное количество тестов для покрытия операторов и покрытия решений: *Discount rate=1;* *Fare = 1000;* *If ((person == “senior citizen”) and (“travel month = January”))* *Bonuspoints = 100+Bonuspoints;* *If (class==”first”)* *discountRate = .5;* *Fare = fare \* discountRate;* |  |
| 1. 1, 1 |  |
| 1. 2, 1 |  |
| 1. 1, 2 |  |
| 1. 2, 2 |  |
| **10.** | **Какие из следующих требований будут проверяться при функциональном тестировании?** |  |
| 1. Работа системы не замедляется при одновременном использовании 100 пользователями |  |
| 1. Система позволяет поменять домашний адрес пользователя |  |
| 1. Система работает при 10000 одновременно залогиненых пользователях |  |
| 1. Система работает в Chrome и Firefox |  |
| **11.** | **Какая информация не обязательно должна присутствовать в отчете о дефекте?** |  |
| 1. Название |  |
| 1. Пошаговый сценарий |  |
| 1. Как исправить дефект |  |
| 1. Кто зарегистрировал |  |
| **12.** | **Что из нижеперечисленного является правдой?** |  |
| 1. Дефект может быть закрыт без исправления |  |
| 1. Заключительный статус дефекта – fixed |  |
| 1. В описании дефекта не может быть информации о тестовом окружении |  |
| **13.** | **Что не может использоваться при составлении тестов?** |  |
| 1. Требования к системе |  |
| 1. Опыт |  |
| 1. Код программы |  |
| 1. Инструкция к системе |  |
| **14.** | **Что не обязательно является правдой при прохождении тест кейса?** |  |
| 1. Если в тест кейсе не указаны конкретные входные данные, то проходиться он должен по позитивному сценарию |  |
| 1. Если в тест кейсе не указаны конкретные входные данные, то нужно учитывать и позитивный и негативный сценарии |  |
| 1. Даже если результаты прохождения тест кейса отличаются от ожидаемых, нужно попытаться закончить прохождение тест кейса |  |
| 1. Не задерживайте прохождение тест кейсов, не отклоняйтесь от шагов в нем |  |
| **15.** | **Что не относится к статическим видам тестирования?** |  |
| 1. Инспекция |  |
| 1. Рецензирование |  |
| 1. Сквозной контроль |  |
| 1. Анализ граничных условий |  |
| **16.** | **Что не относится к методам тестирования черного ящика?** |  |
| 1. Эквивалентное разбиение |  |
| 1. Тестирование таблицы переходов |  |
| 1. Исследовательское тестирование |  |
| **17.** | **Задача: Если зарплата работника до 4000 гривен, он не облагается налогом.**  **Следующие 1500 гривен облагаются налогом 10%.**  **Следующие 28000 – 22%.**  **Все, что больше, облагается налогом 40%.**  **Какие из перечисленных чисел попадают в разные классы эквивалентности?** |  |
| 1. 4000; 5000; 5500 |  |
| 1. 32001; 34000; 36500 |  |
| 1. 28000; 28001; 32001 |  |
| 1. 4000; 4200; 5600 |  |
| **18.** | Дана следующая таблица решений:   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | Правило1 | Правило2 | Правило3 | Правило4 | | Условия: |  |  |  |  | | Гражданин UK? | нет | да | да | да | | Возраст 18-55 лет? | не важно | нет | да | да | | Курит? | не важно | не важно | нет | да | | Действия: |  |  |  |  | | Страховать клиента? | нет | нет | да | да | | Предложить скидку 10%? | нет | нет | да | нет |   Что будет ожидаемыми результатами для каждого из тест кейсов? A. Фред живет в Лондоне, ему 32 года, он курит B. Михаилу 65 лет, он живет в Париже и не курит |  |
| 1. A – страховать, 10% скидка, B – страховать, нет скидки |  |
| 1. A – не страховать, B – не страховать |  |
| 1. A – страховать, нет скидки, B – не страховать |  |
| 1. A - страховать, нет скидки, B- не страховать |  |
| **19.** | **Что для семантического версионирования не является правдой?**  **Учитывая номер версии МАЖОРНАЯ.МИНОРНАЯ.ПАТЧ, следует увеличивать:** |  |
| 1. МАЖОРНУЮ версию, когда сделаны обратно несовместимые изменения API |  |
| 1. МИНОРНУЮ версию, когда вы добавляете новый функционал, не нарушая обратной совместимости |  |
| 1. ПАТЧ-версию, когда вы делаете обратно совместимые исправления |  |
| 1. ПАТЧ-версию, когда вы добавляете новый функционал, не нарушая обратной совместимости |  |
| **20.** | **Branch – это:** |  |
| 1. направление разработки, независимое от других |  |
| 1. основная версия |  |
| 1. рабочая (локальная) копия документов |  |
| 1. извлечение документа из хранилища и создание рабочей копии |  |
| **21.** | **Кто обычно осуществляет debugging?** |  |
| 1. Аналитики |  |
| 1. Разработчики |  |
| 1. Тестировщики |  |
| **22.** | **Что является главным кандидатом для покрытия unit-тестами для тестировщиков?** |  |
| 1. Web-сервиса |  |
| 1. Открытый API |  |
| 1. Новые добавленные классы, методы и т.д. |  |
| 1. Часто меняющиеся места в коде |  |
| **23.** | **Что является главным кандидатом для покрытия автотестами?** |  |
| 1. Части ПО с часто меняющимся интерфейсом |  |
| 1. Регрессионные тесты |  |
| 1. Части ПО в редко меняющимся интерфейсом |  |
| **24.** | **Регрессионное тестирование должно проводится:** |  |
| 1. Раз в неделю |  |
| 1. Так часто, как возможно |  |
| 1. Когда project manager сказал |  |
| 1. Когда меняется окружение |  |
| **25.** | **Тест план содержит все из нижеперечисленного, кроме:** |  |
| 1. Что должно быть протестировано |  |
| 1. Что не должно быть протестировано |  |
| 1. Спецификации для тестов |  |
| 1. Задачи по тестированию |  |