

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Косенко Сергей Михайлович

Должность: ректор

Дата подписания: 24.06.2026 06:57:07

Уникальный идентификатор:

e3a68f3eaa1e62674b54f4998099d3d6bdfcf836

Тестовое задание для диагностического тестирования по дисциплине

Инструментальные средства информационных систем, 7 семестр

| | |
|-----------------------------|---|
| Код, направление подготовки | 09.03.02 Информационные системы и технологии |
| Направленность (профиль) | Безопасность информационных систем и технологий |
| Форма обучения | Очная |
| Кафедра разработчик | ИВТ |
| Выпускающая кафедра | ИВТ |

| Проверяемая компетенция | № | Задание | Варианты ответов | Тип сложности вопроса |
|---|---|--|--|-----------------------|
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-5.1 ПК-5.2 ПК- 5.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 | 1 | Сколько систем контроля версий существует? | 1) Только Git 2) Git и SVN 3) Git и CVS 4) Много | низкий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 2 | Что такое GitHub? | 1) Программа для работы с Git 2) Драйвер для Git 3) Веб-сервис для хостинга IT-проектов и их совместной разработки, основанный на Git 4) UI для работы с локальной версией Git | низкий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 3 | Что такое репозиторий Git? | 1) Любая директория/папка в моей ОС 2) Любая папка, находящаяся внутри Git 3) Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов, и хранилище, содержащее собственно файлы 4) Папка .git/ и все входящие в нее | низкий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 4 | Что делает команда git status? | 1. Показывает состояние проекта: кол-во untracked, deleted, new и прочих файлов, количество коммитов, на которое отличается локальная версия репозитория от удаленного и так далее 2. Показывает имя и email нашего пользователя, а также является ли он авторизованным в системе GitHub или нет | низкий |

| | | | | |
|---|----|---|--|---------|
| | | | 3. Показывает место, занимаемое репозиторием на жестком диске и кол-во выделенного под репозиторий месте 4. Такой команды нет, есть только команда <code>git show</code> | |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 5 | Что делает команда <code>git add</code> ? | 1) Создает файл с указанным именем и сразу добавляет его в Git 2) Добавляет локальный файл в удаленный репозиторий так, чтобы другие участники проекта могли его видеть 3) Это алиас/синоним команды <code>git commit</code> 4) Начинает отслеживать указанный файл или файлы | низкий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 6 | Что означает статус файла <code>untracked</code> в выводе команды <code>git status</code> ? | 1) Что система Git не отслеживает этот файл 2) Что файл был удален из Git 3) Что файл находится вне репозитория Git 4) Что файл добавлен в <code>.gitignore</code> | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 7 | Что означает статус файла <code>new</code> в выводе команды <code>git status</code> ? | 1) Что файл только что был создан и еще не отслеживается системой Git 2) Что файл только начал отслеживаться Git и пока не имеет истории 3) Что файл удаляли из Git и потом восстановили командой <code>git return</code> 4) Такого статуса нет, есть только статус <code>deleted file</code> | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 8 | Что означает статус файла <code>modified</code> в выводе команды <code>git status</code> ? | 1) Что файл имеет историю в системе Git и был изменен относительно его последнего состояния 2) Такого статуса нет, есть только статусы <code>new</code> и <code>deleted</code> 3) Этот статус виден только командой <code>git ignore</code> и означает, что файл перестал отслеживаться системой Git 4) Статус означает, что файл добавлен в коммит | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 9 | Что такое коммит? | 1) Это единица состояния проекта в Git 2) Это результат вывода команды <code>git diff</code> 3) Это обобщающее название одного из статусов файла в выводе <code>git status</code> : <code>untracked</code> , <code>new</code> , <code>deleted</code> или <code>modified</code> 4) Это слово ничего не означает, его ввели только для того, чтобы путать новичков | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 10 | Как сделать коммит? | 1) Всего лишь набрать команду <code>git commit</code> в любой момент времени 2) Сделать изменения в файлах и перечислить их после <code>git commit</code> . Например так: <code>git commit a.file, b.file</code> 3) Сделать изменения, собрать эти изменения командой <code>"git add"</code> или <code>"git commit -a"</code> и указать коммит-сообщение после ключа <code>"-m"</code> | средний |

| | | | | |
|---|----|--|--|---------|
| | | | 4) Нельзя сделать коммит, ведь такого понятия не существует | |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 11 | Что такое ветка в репозитории Git? | 1) Это то же самое, что и коммит 2) Это минимум два коммита с одинаковым коммит-сообщением 3) Это разные пути развития проекта, по сути разные последовательности коммитов 4) Это механизм изменения конкретного файла | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 12 | Чем отличается master и origin master? | 1) Это просто два разных названия одной ветки 2) master принадлежит локальному репозиторию, а origin master - удаленному 3) Это две разные ветки локального репозитория 4) Ветки origin master не существует | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 13 | Чем отличаются команды "git push" и "git pull"? | 1) Это алиасы 2) Команды "git pull" не существует, а команда "git push" нужна, чтобы выложить изменения в удаленный репозиторий 3) Команды "git push" не существует, а команда "git pull" нужна, чтобы стянуть изменения из удаленного репозитория 4) команда "git pull" нужна, чтобы стянуть изменения из удаленного репозитория, а команда "git push" нужна, чтобы выложить изменения в удаленный репозиторий | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 14 | Что делает команда git log? | 1) Пишет указанный после файл в лог 2) Такой команды нет, есть только команда git look 3) Показывает историю коммитов 4) Удаляет файл из репозитория | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 15 | Что делает команда git show? | 1) Показывает изменения, сделанные в указанном коммите 2) Показывает содержимое файла 3) Показывает состояние проекта 4) Показывает время | средний |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 | 16 | Как узнать, кто автор строки в файле, используя систему Git? | 1) Команда git show --author 2) Команда git commit --author 3) Команда git blame 4) Команда git status | высокий |

| | | | | |
|---|----|---|--|---------|
| ПК-5.1 ПК-5.2 ПК- 5.3 | | | | |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 17 | Как узнать, какие изменения мы сделали локально относительно последнего состояния нашего репозитория? | <ol style="list-style-type: none"> 1) Команда git show 2) Команда git diff 3) Команда git izmeneniya 4) Команда git commit | высокий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 18 | Как отменить действие команды "git add" на файл? | <ol style="list-style-type: none"> 1) Команда git abort 2) Команда git stash 3) Команда git not-add 4) Команда git reset | высокий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 19 | Как решить конфликт в Git? | <ol style="list-style-type: none"> 1) Руками поправить изменения там, где Git не смог это сделать автоматически и затем собрать все в коммит и запустить 2) Никак, придется создавать репозиторий заново 3) Выполнить команду git commit merge please 4) Удалить файл, для которого Git не знает, как смиржить изменения | высокий |
| ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3 | 20 | Как привести измененный файл в начальное состояние (до изменения)? | <ol style="list-style-type: none"> 1) Команда git abort path/to/file 2) Команда git checkout path/to/file 3) Команда git pull path/to/file 4) Команда git commit path/to/file | высокий |