

Документ подписан электронной подписью  
 Информация о владельце:  
 ФИО: Косенок Сергей Михайлович  
 Должность: ректор  
 Дата подписания: 22.06.2026 12:41:44  
 Уникальный программный ключ:  
 e3a68f3eaa1e02674b5414998099d3d6bfdcf836

**Тестовое задание для диагностического тестирования по дисциплине:**

**Инструментальные средства ИС, 7 семестр**

Код, направление подготовки	09.03.02
Направленность (профиль)	Информационные системы и технологии
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-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	Сколько систем контроля версий существует?	<ol style="list-style-type: none"> <li>1) Только Git</li> <li>2) Git и SVN</li> <li>3) Git и CVS</li> <li>4) Много</li> </ol>	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	Что такое GitHub?	<ol style="list-style-type: none"> <li>1) Программа для работы с Git</li> <li>2) Драйвер для Git</li> <li>3) Веб-сервис для хостинга IT-проектов и их совместной разработки, основанный на Git</li> <li>4) UI для работы локальной версией Git</li> </ol>	низкий

<p>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</p>	<p>Что такое репозиторий Git?</p>	<p>1) Любая директория/папка в моей ОС</p> <p>2) Любая папка, находящаяся внутри Git</p> <p>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</p> <p>4) Папка .git/ и все входящие в нее</p>	<p>низкий</p>
<p>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</p>	<p>Что делает команда git status?</p>	<p>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</p> <p>2) Показывает имя и email на шего пользователя, а также является ли он авторизованным</p>	<p>низкий</p>

		<p>в системе GitHub или нет</p> <p>3) Показывает место, занимаемое репозиторием на жестком диске и кол-во выделенного под репозиторий месте</p> <p>4) Такой команды нет, есть только команда git show</p>	
<p>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</p>	<p>Что делает команда git add?</p>	<p>1) Создает файл с указанным именем и сразу добавляет его в Git</p> <p>2) Добавляет локальный файл в удаленный репозиторий так, чтобы другие участники проекта могли его видеть</p> <p>3) Это алиас/синоним команды git commit</p> <p>4) Начинает отслеживать указанный файл или файлы</p>	<p>низкий</p>
<p>ПК-1.1 ПК- 1.2 ПК-1.3</p>	<p>Что означает статус файла untracked в выводе команды git status?</p>	<p>1) Что система Git не отслеживает этот файл</p>	<p>средний</p>



<p>ПК -11.2 ПК- 11.3 ПК- 14.1 ПК- 14.2 ПК- 14.3 ПК-5.1 ПК-5.2 ПК- 5.3</p>		<p>2) Что файл был удален из Git</p> <p>3) Что файл находится вне репозитория Git</p> <p>4) Что файл добавлен в .gitignore</p>	
<p>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</p>	<p>Что означает статус файла new в выводе команды git status?</p>	<p>1) Что файл только что был создан и еще не отслеживается системой Git</p> <p>2) Что файл только начал отслеживаться Git и пока не имеет истории</p> <p>3) Что файл удаляли из Git и потом восстановили командой git return</p> <p>4) Такого статуса нет, есть только статус deleted file</p>	<p>средний</p>
<p>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</p>	<p>Что означает статус файла modified в выводе команды git status?</p>	<p>1) Что файл имеет историю в системе Git и был изменен относительно его последнего состояния</p> <p>2) Такого статуса нет, есть только статус new и deleted</p> <p>3) Этот статус виден только командой git ignore и означает, что файл перестал</p>	<p>средний</p>

		<p>отслеживаться системой Git</p> <p>4) Статус означает, что файл добавлен в коммит</p>	
<p>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</p>	<p>Что такое коммит?</p>	<p>1) Это единица состояния проекта в Git</p> <p>2) Это результат вывода команды git diff</p> <p>3) Это обобщающее название одного из статусов файла в выводе git status: untracked, new, deleted или modified</p> <p>4) Это слово ничего не означает, его ввели только для того, чтобы путать новичков</p>	<p>средний</p>
<p>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</p>	<p>Как сделать коммит?</p>	<p>1) Всего лишь набрать команду git commit в любой момент времени</p> <p>2) Сделать изменения в файлах и перечислить их после git commit. Например так: git commit a.file, b.file</p>	<p>средний</p>

		<p>3) Сделать изменения, собрать эти изменения командой "git add" или "git commit -a" и указать коммит-сообщение после ключа "-m"</p> <p>4) Нельзя сделать коммит, ведь такого понятия не существует</p>	
<p>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</p>	<p>Что такое ветка в репозитории Git?</p>	<p>1) Это то же самое, что и коммит</p> <p>2) Это минимум два коммита с одинаковым коммит-сообщением</p> <p>3) Это разные пути развития проекта, по сути разные последовательности коммитов</p> <p>4) Это механизм изменения конкретного файла</p>	<p>средний</p>
<p>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</p>	<p>Чем отличается master и origin master?</p>	<p>1) Это просто два разных названия одной ветки</p> <p>2) master принадлежит локальному репозиторию, а origin master - удаленному</p> <p>3) Это две разные ветки</p>	<p>средний</p>

		<p>локального репозитория</p> <p>4) Ветки origin master не существует</p>	
<p>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</p>	<p>Чем отличаются команды "git push" и "git pull"?</p>	<p>1) Это алиасы</p> <p>2) Команды "git pull" не существует, а команда "git push" нужна, чтобы выложить изменения в удаленный репозиторий</p> <p>3) Команды "git push" не существует, а команда "git pull" нужна, чтобы стянуть изменения из удаленного репозитория</p> <p>4) команда "git pull" нужна, чтобы стянуть изменения из удаленного репозитория, а команда "git push" нужна, чтобы выложить изменения в удаленный репозиторий</p>	<p>средний</p>
<p>ПК-1.1 ПК- 1.2 ПК-1.3 ПК-11.1 ПК -11.2</p>	<p>Что делает команда git log?</p>	<p>1) Пишет указанный после файл в лог</p>	<p>средний</p>

<p>ПК- 11.3  ПК- 14.1  ПК- 14.2  ПК- 14.3  ПК-5.1  ПК-5.2  ПК- 5.3</p>		<p>2) Такой команды нет, есть только команда git look  3) Показывает историю коммитов  4) Удаляет файл из репозитория</p>	
<p>ПК-1.1  ПК- 1.2  ПК-1.3  ПК-11.1  ПК -11.2</p> <p>ПК- 11.3  ПК- 14.1  ПК- 14.2  ПК- 14.3  ПК-5.1  ПК-5.2  ПК- 5.3</p>	<p>Что делает команда git show?</p>	<p>1) Показывает изменения, сделанные в указанном коммите  2) Показывает содержимое файла  3) Показывает состояние проекта  4) Показывает время</p>	<p>средний</p>
<p>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</p>	<p>Как узнать, кто автор строки в файле, используя систему Git?</p>	<p>1) Команда git show --author  2) Команда git commit --author  3) Команда git blame  4) Команда git status</p>	<p>высокий</p>
<p>ПК-1.1  ПК- 1.2</p> <p>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</p>	<p>Как узнать, какие изменения мы сделали локально относительно последнего состояния нашего репозитория?</p>	<p>1) Команда git show  2) Команда git diff  3) Команда git izmeneniya  4) Команда git commit</p>	<p>высокий</p>

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	Как отменить действие команды "git add" на файл?	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	Как решить конфликт в Git?	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 смирить 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	Как привести измененный файл в начальное состояние (до изменения)?	1) Команда git abort path/to/file  2) Команда git checkout path/to/file  3) Команда git pull path/to/file	высокий

ПК-5.2 ПК- 5.3		4) Команда git commit path/to/file	
-------------------	--	---------------------------------------	--