МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Методические указания

по организации самостоятельной работы обучающихся
по дисциплине «**ПРОЕКТИРОВАНИЕ, ВНЕДРЕНИЕ, СОПРОВОЖДЕНИЕ, НАСТРОЙКА И ЭКСПЛУАТАЦИЯ ИНФОРМАЦИОННЫХ СИСТЕМ**»
для студентов направления подготовки /специальности

09.03.02 Информационные системы и технологии

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Содержание

ВВЕДЕНИЕ

Самостоятельная работа студента (СРС) наряду с аудиторной представляет одну из форм учебного процесса и является существенной его частью. СРС – это планируемая работа студентов, выполняемая по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

СРС предназначена не только для овладения каждой дисциплиной, но и для формирования навыков самостоятельной работы вообще, в учебной, научной, профессиональной деятельности, способности принимать на себя ответственность, самостоятельно решить проблему, находить конструктивные решения.

1. Общая характеристика самостоятельной работы при изучении дисциплины «Методы и средства проектирования информационных систем»

Самостоятельная работа предусматривает следующие виды: Изучение литературы по темам, вынесенным на самостоятельную работу, Подготовка к лабораторным работам (решение разноуровневых задач).

Цель самостоятельной работы:

1. углублять и расширять профессиональные знания;
2. формировать у студентов интерес к учебно-познавательной деятельности;
3. научить студентов овладевать приемами процесса познания.

Задачи самостоятельной работы:

1. развивать у студентов самостоятельность, активность, ответственность;
2. развивать познавательные способности будущих специалистов.

3. Контрольные точки и виды отчетности по ним

№ п/п	Вид деятельности студентов	Сроки выполнения	Количество баллов
1.	Собеседование по темам	5-ая неделя	15

2.	ЛЗ (решение разноуровневых задач)	7-ая неделя	15
3.	ЛЗ (решение разноуровневых задач)	12 –ая неделя	25
	Итого за 5 семестр		**55**

4. Методические указания по изучению теоретического материала

4.1 Вид самостоятельной работы: самостоятельное изучение литературы

Изучать учебную дисциплину рекомендуется по темам, предварительно ознакомившись с содержанием каждой из них в программе дисциплины. При теоретическом изучении дисциплины студент должен пользоваться соответствующей литературой. Примерный перечень литературы приведен в рабочей программе

Для более полного освоения учебного материала студентам читаются лекции по важнейшим разделам и темам учебной дисциплины. На лекциях излагаются и детально рассматриваются наиболее важные вопросы, составляющие теоретический и практический фундамент дисциплины. В процессе изучения учебной дисциплины студент должен выполнить контрольную работу, целью которой является приобретение практических навыков нормирования и оценки эффективности технологических решений.

Итоговый продукт: Конспект статей

Средства и технологии оценки: Собеседование

Критерии оценивания: Оценка «отлично» выставляется студенту, если в полном объеме изучен курс данной дисциплины и выполнены практические задания

Оценка «хорошо» выставляется студенту, если достаточно полно изучен курс данной дисциплины и выполнены практические задания

Оценка «удовлетворительно» выставляется студенту, недостаточно если полно изучен курс данной дисциплины и выполнены практические задания

Оценка «неудовлетворительно» выставляется студенту, если отсутствуют знания и практические навыки по данной дисциплине

Темы для самостоятельного изучения

1. Основные понятия начертательной геометрии и инженерной графики. Задание точки, прямой, плоскости и многогранников на комплексном чертеже, многогранники.
2. Простые геометрические построения. Построение сопряжений..
3. Аксонометрические проекции. Построение ортогональных и аксонометрических проекций многогранников и тел вращения..

4.2 Вид самостоятельной работы:Подготовка к лабораторным работам (решение разноуровневых задач)

Итоговый продукт: Лабораторная работа

Средства и технологии оценки: Отчет письменный

Критерии оценивания: Оценка «отлично» выставляется студенту, если в полном объеме изучен курс данной дисциплины и выполнены лабораторные задания

Оценка «хорошо» выставляется студенту, если достаточно полно изучен курс данной дисциплины и выполнены лабораторные задания

Оценка «удовлетворительно» выставляется студенту, недостаточно, если полно изучен курс данной дисциплины и выполнены лабораторные задания

Оценка «неудовлетворительно» выставляется студенту, если отсутствуют знания и практические навыки по данной дисциплине

Задания для лабораторных работ

Лабораторная работа №1 «Первоначальная настройка git»

Тема: Первоначальная настройка git. Инициализация каталога. Состоянияфалов в git. Первый коммит.

Цель работы: провести первоначальную настройку системы контроля версии git, после установки инициализировать каталог для работы, разобраться с существующими состояниями файлов в git, сделать первый коммит.

В состав git'a входит утилита **gitconfig**, которая позволяет просматривать иустанавливать параметры, контролирующие все аспекты работы git'a и его внешнийвид.

Первое, что необходимо сделать после установки git'a, — указать имя и адресэлектронной почты. Это важно, потому что каждый коммит в git'e содержит эту информацию, и она включена в коммиты, передаваемые разработчиками, и не можетбыть далее изменена.

Для того чтобы начать использовать git для существующего проекта, необходимо перейти в проектный каталог и в командной строке ввести **gitinit**.Эта команда создаёт в текущем каталоге новый подкаталог с именем .git содержащий все необходимые файлы репозитория — основу git-репозитория. На этомэтапе проект ещё не находится под версионным контролем. Данная команда инициализирует возможность работы с git, но не вносит файлы под контроль.

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Зайти в папку T://{Номер группы} и в ней создать папку соответствующую инициалам студента на английском языке. Например, для студента ИвановПетр Петрович, папка будет иметь имя IPP.
3. Провести инициализацию репозитория в созданной папке. Для этого, открыть программу **GitBash**, перейти в созданную папку (для перемещения используется команда **cd T://{Номер группы}/{Инициалы}**).
4. Установить настройки имени и e-mail’a, не используя опцию **--global.**
5. Создать в папке файл my_first_file.txt и проиндексировать его.
6. Сделать первый коммит.
7. Открыть файл my_first_file.txt и добавить в него строчку «testrow». Проиндексировать изменения.
8. Создать новый файл my_second_file.txt. Проиндексировать изменения.
9. Сделать второй коммит.
10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

Лабораторная работа № 2 «Игнорирование, сравнение, удаление иперемещение файлов»

Тема: Игнорирование файлов. Сравнение изменений. Удаление и перемещениефайлов.

Цель работы: научиться исключать файлы, которые нет необходимостивести в системе контроля версий. Получить практические навыки сравнения проделанных изменений в файлах.

Зачастую, имеется группа файлов, которые не только нет необходимости автоматически добавлять в репозиторий, но и видеть в списках неотслеживаемых. Ктаким файлам обычно относятся автоматически генерируемые файлы (различныелоги, результаты сборки программ и т.п.). В таком случае, необходимо создать файл.*gitignore*с перечислением шаблонов соответствующих таким файлам.К шаблонам в файле .gitignore применяются следующие правила:

– Пустые строки, а также строки, начинающиеся с # (символ комментария), игнорируются.
– Можно использовать стандартные glob шаблоны.
– Можно заканчивать шаблон символом слэша (/) для указания каталога.
– Можно инвертировать шаблон, использовав восклицательный знак (!) вкачестве первого символа.

Glob-шаблоны представляют собой упрощённые регулярные выражения используемые командными интерпретаторами. Символ * соответствует 0 или болеесимволам; последовательность [abc] — любому символу из указанных в скобках (вданном примере a, b или c); знак вопроса (?) соответствует одному символу; [0-9]соответствует любому символу из интервала (в данном случае от 0 до 9).

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием на первой лабораторнойработе.
3. Создать папку temp в своем репозитории.
4. Создать папку log и добавить в нее 2 файла: main.html и some.tmp.
5. Создать файл .gitignore и добавить в игнорирование папку temp и файлыс расширением .tmp из папки log.
6. Закоммитить добавление файла .gitignore.
7. Внести изменения в файл my_first_file.txt, добавив строчку “rowtoindex», проиндексировать данные изменения. Еще раз внести изменения в файл, добавив строчку “rowno index».
8. Посмотреть индексированные и неиндексированные изменения используя команду **git diff.**
9. Удалить файл my_first_file.txt, зафиксировать данное удаление.
10. Переименовать файл my_second_file.txt в my_first_file.txt, зафиксироватьизменение.
11. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопрос

Лабораторная работа №3 «Просмотр истории коммитов»

Тема: Просмотр истории коммитов, команда gitlog.

Цель работы: освоить механизм работы с командой gitlog для полученияинформации об истории коммитов.

После того как будет создано несколько коммитов, вероятнее всего появитсянеобходимость просмотреть, что же происходило с этим репозиторием. Наиболеепростой и в то же время мощный инструмент для этого — команда **gitlog**.По умолчанию, без аргументов, **gitlog**выводит список коммитов созданных вданном репозитории в обратном хронологическом порядке. То есть самые последние коммиты показываются первыми.Один из наиболее полезных параметров — это **-p**, который показывает дельту(разницу/diff), привнесенную каждым коммитом. Также можно использовать **-2**, чтоограничит вывод до 2-х последних записей.

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Изучить возможности команды **gitlog**, выполнить различные вариантывывода информации и ее отбора.
4. Выполнить задание согласно варианту.
5. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

Варианты

Номер	**Задание**
1	Вывести коммиты, автором которых являетесь Вы, за последний месяц.
2	Вывести все коммиты в формате: короткий хеш, автор, комментарий.
3	Вывести все коммиты, в сообщении которых присутствует слово my.
4	Вывести все коммиты за текущий месяц с информацией о том, какие файлы были изменены.
5	Вывести информацию о первом коммите в системе, с выводом дельты (diff)
6	Вывести коммиты сделанные за последний месяц назад, но исключаяпоследнюю неделю.
7	Вывести информацию о коммитах в формате: автор, дата, список измененных файлов.
8	Вывести коммиты, автором которых являетесь Вы, с выводом дельты(diff).
9	Вывести коммиты, в которых происходили изменения файламy_first_file.txt, за последние 2 недели.
10	Вывести последние 3 коммита в формате: автор, комментарий.
11	Вывести информацию о первом коммите в формате: дата, автор, комментарий, а также список измененных файлов.
12	Вывести коммиты, автором которых являетесь Вы, со списком измененных файлов.
13	Вывести все коммиты за текущий месяц в формате: сокращенный

	хеш,дата, комментарий.
14	Вывести все коммиты в формате: e-mail автора, дата коммита, хеши родительских коммитов.
15	Вывести коммиты, в которых происходили изменения файламy_first_file.txt.

Лабораторная работа №4 «Отмена изменений. Работа с метками»

Тема: Отмена внесенных изменений. Работа с метками.

Цель работы: научиться отменять сделанные изменения, работать с метками.

Одна из типичных отмен происходит тогда, когда коммит сделан слишком рано, например, не были добавлены какие-либо файлы, или перепутан комментарий ккоммиту. Если необходимо сделать этот коммит ещё раз, можно выполнить **gitcommit**с опцией **–amend.**

Эта команда берёт индекс и использует его для коммита. Если после последнего коммита не было никаких изменений (например, приведенная команда былазапущена сразу после предыдущего коммита), то состояние проекта будет абсолютно таким же и всё, что изменится, это комментарий к коммиту.Git использует два основных типа меток: легковесные и аннотированные. Легковесная метка — это что-то весьма похожее на ветку, которая не меняется — этопросто указатель на определённый коммит. А вот аннотированные метки хранятся вбазе данных Git'a как полноценные объекты. Они имеют контрольную сумму, содержат имя поставившего метку, e-mail и дату, имеют комментарий и могут бытьподписаны и проверены с помощью GNU PrivacyGuard (GPG). Обычно рекомендуется создавать аннотированные метки, чтобы иметь всю перечисленную информацию; но если необходимо сделать временную метку или по какой-то причине нетнеобходимости сохранять остальную информацию, то для этого годятся и легковесные метки.

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Создать три файла: 1.txt, 2.txt, 3.txt.
4. Проиндексировать первый файл и сделать коммит с комментарием “add1.txt file».
5. Проиндексировать второй и третий файлы.
6. Удалить из индекса второй файл.
7. Перезаписать уже сделанный коммит с новым комментарием “add 1.txtand 3.txt»
8. Создать аннотированную метку с названием v0.01.
9. Создать легковесную ветку указывающую на первый коммит в репозитории.
10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

Лабораторная работа №5 «Ветвление. Конфликты»

Тема: Работа с ветками, решение конфликтов.

Цель работы: научиться создавать ветки, перемещаться по ним, объединять и удалять их. Решать конфликты слияния.

Ветка в git'e — это просто легковесный подвижный указатель на один изкоммитов. Ветка по умолчанию в git'e называется **master**. Когда происходит создание коммита на начальном этапе, доступна ветка **master**, указывающая на последний сделанный коммит. При каждом новом коммите она сдвигается вперёд автоматически.Для того чтобы создать новую ветку используется команда **gitbranch**.Эта команда создаст новый указатель на тот самый коммит, на котором сейчаснаходится git.

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Создать новую ветку my_first_branch.
4. Перейти на ветку и создать новый файл in_branch.txt, закоммитить изменения.
5. Вернуться на ветку master.
6. Создать и сразу перейти на ветку new_branch.
7. Сделать изменения в файле 1.txt, добавить строчку “newrowin 1.txt file»,закоммитить изменения.
8. Перейти на ветку master и слить ветки master и my_first_branch, послечего слить ветки master и new_branch.
9. Удалитьветкиmy_first_branchи new_branch.
10. Создать ветки branch_1 и branch_2.
11. Перейти на ветку branch_1 и изменить файл 1.txt, удалить все содержимое и добавить текст “fixin 1.txt», изменить файл 3.txt, удалить все содержимое идобавить текст “fixin 3.txt», закоммитить изменения.
12. Перейти на ветку branch_2 и также изменить файл 1.txt, удалить все содержимое и добавить текст “Myfixin 1.txt», изменить файл 3.txt, удалить все содержимое и добавить текст “Myfixin 3.txt», закоммитить изменения.
13. Слить изменения ветки branch_2 в ветку branch_1.
14. Решить конфликт файла 1.txt в ручном режиме, а конфликт 3.txt используя команду gitmergetool с утилитой Meld.
15. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

Лабораторная работа №6 «Прятанье»

Тема: Механизм прятанья.

Цель работы: научиться использовать механизм прятанья, а также расширить знания в управлении веток.

Часто возникает такая ситуация, что пока идет работа над частью своего проекта, всё находится в беспорядочном состоянии, а нужно переключить ветки, чтобынемного поработать над чем-то другим. Проблема в том, что делать коммит с наполовину доделанной работой только для того, чтобы позже можно было вернуться вэто же состояние не хотелось бы. Ответ на эту проблему — команда **gitstash**.Прятанье поглощает грязное состояние рабочего каталога, то есть изменённыеотслеживаемые файлы и изменения в индексе, и сохраняет их в стек незавершённых
изменений, которые потом в любое время можно снова применить.

Порядок выполнения работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Проверить какие ветки слиты с веткой master, а какие нет.
4. Удалить все ветки слитые с master.
5. Создать новую ветку work и перейти в нее.
6. Изменить файл 1.txt.
7. Спрятать данные изменения.
8. Развернуть обратно данные изменения с опцией --index.
9. Развернуть спрятанные изменения в новую ветку.
10. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы.

Лабораторная работа №7 «Работа с удаленным репозиторием»

Тема: Работа с удаленным репозиторием. Github.com.

Цель работы: научиться работать с удаленным репозиторием, использовать платформу github.com.

Если необходимо получить копию существующего репозиторияGit, например, проекта, в котором разработчик планирует поучаствовать, то необходимо использовать команду **gitclone**. Каждая версия каждого файла из истории проекта забирается (*pulled*) с сервера, когда выполняется команда **gitclone**. Фактически, еслисерверный диск выйдет из строя, можно использовать любой из клонов на любом изклиентов, для того чтобы вернуть сервер в то состояние, в котором он находился вмомент клонирования.Клонирование репозитория осуществляется командой **gitclone [url].**

Порядок выполнения практической работы

1. Изучить теоретическую часть работы.
2. Продолжить работу с созданным репозиторием.
3. Пройти регистрацию на сайте github.com.
4. Настроить доступ к github по SSH.
5. Склонироватьрепозиторийgit@github.com:mavose/tusur_{номер группы}.git.
6. Продемонстрировать преподавателю ход работы, ответить на уточняющие вопросы

Список литературы

Основная литература

1. Рашевская, М. А. Компьютерные технологии в дизайне среды : [учеб. пособие] / М.А. Рашевская. - М. : ФОРУМ, 2011. - 304 с. - Прил.: с. 272-296. - ISBN 978-5-91134-227-2
2. Орлов, А. AutoCAD 2014 / А. Орлов. - СПб. : Питер, 2014. - 384 с. : ил. - Прил.: с. 382. - ISBN 978-5-496-00761-0
3. Инженерная и компьютерная графика : лабораторный практикум / авт.-сост. Т.И. Дровосекова ; Сев.-Кав. федер. ун-т. - Ставрополь : СКФУ, 2014. - 2015. - Библиогр.: с. 159
4. Семенова, Н.В. Инженерная графика : учебное пособие / Н.В. Семенова, Л.В. Баранова. - Екатеринбург : Издательство Уральского университета, 2014. - 89 с. : схем., табл., ил. - Библиогр.: с. 71. - ISBN 978-5-7996-1099-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=275945
5. Конакова, И.П. Инженерная и компьютерная графика : учебное пособие / И.П. Конакова, И.И. Пирогова ; Министерство образования и науки Российской Федерации, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина. - Екатеринбург : Издательство Уральского университета, 2014. - 91 с. : схем., ил. - Библиогр.: с. 59. - ISBN 978-5-7996-1312-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=275737

Дополнительная литература:

1. Полещук, Н. Н. Самоучитель AutoCAD 2013 / Н.Н. Полещук. - СПб. : БХВ-Петербург, 2013. - 464 с. : ил. - (Самоучитель). - Прил.: с. 136-444. - Библиогр.: с. 445. - ISBN 978-5-9775-0889-6
2. Берлинер, Э. М. САПР в машиностроении : учебник для вузов / Э. М. Берлинер, О. В. Таратынов. – Москва : Форум, 2014. – 448 с.
3. Гумерова, Г.Х. Основы компьютерной графики: учебное пособие / Г.Х. Гумерова ; Министерство образования и науки России, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Казанский национальный исследовательский технологический университет». - Казань : Издательство КНИТУ, 2013. - 87 с. : ил., табл. - Библиогр. в кн. - ISBN 978-5-7882-1459-7 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=258794

Интернет-ресурсы

Для проработки теоретического материала и выполнения самостоятельных работ рекомендуется использовать следующие Интернет-ресурсы:

1. http://www.biblioclub.ru/ - электронная библиотека
2. http://www.uts-edu.ru/ - «Электронные курсы»

Программное обеспечение

- Windows 7.
- Github