

**Министерство просвещения РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
Ульяновский государственный педагогический университет
им. И.Н. Ульянова
Факультет образовательных технологий
и непрерывного образования
Кафедра методики естественнонаучного образования и
информационных технологий**

Современные педагогические технологии в преподавании предметов естественно- математического цикла

**УЛЬЯНОВСК
2020**

**ББК 88.6
О-64**

**Печатается по решению ученого совета
факультета образовательных технологий и
непрерывного образования ФГБОУ ВО
«УлГПУ им. И.Н. Ульянова»**

Редколлегия: Данилов С.В. – к.пс.н., доцент, директор центра образовательных перспектив и инноваций.
Мукина О.Г. – к.ф.н., доцент кафедры методики естественнонаучного образования и информационных технологий УлГПУ им. И.Н. Ульянова, научный редактор.

**Современные педагогические технологии в преподавании предметов
естественно-математического цикла [Текст] : сборник научных трудов. –
Ульяновск : ФГБОУ ВО «УлГПУ им. И.Н. Ульянова», 2020. – 126 с.**

В сборнике научных трудов представлены материалы, посвящённые выявлению и обсуждению актуальных проблем, перспектив и тенденций в преподавании предметов естественно-математического цикла в школе, в вузе, исходя из контекста современной образовательной политики.

В сборнике представлены: перспективы и тенденции преподавания предметов естественно-математического цикла; информационно-коммуникационные технологии, электронные образовательные ресурсы для предметов естественно-математического цикла; методическое обеспечение предметов естественно-математического цикла; использование современных педагогических технологий в преподавании предметов естественно-математического цикла; дистанционное обучение предметам естественно-математического цикла; педагогические и методические проблемы преподавания предметов естественно-математического цикла и способы их разрешения.

Для профессорско-преподавательского состава, научных сотрудников, методистов институтов постдипломного педагогического образования, руководителей и специалистов органов управления образованием, образовательных организаций, педагогов.

**ББК 88.6
О-64**

© ФГБОУ ВО «УлГПУ им. И.Н. Ульянова»

СОДЕРЖАНИЕ

Информационные технологии в обучении

Кукаев Н.А., Жадаев А.Ю., Новик И.Р. Дистанционное обучение химии во время пандемии	6
Почаева Н.Д. Использование интернет - ресурсов на уроке географии	9
Басырова Э.И. Виртуальная доска IDROO как помощник в дистанционном обучении	13
Биккулова Р.К., Байбикова Д.Р. Разработка электронного образовательного курса в помощь дистанционному образованию	16
Бикметова В.П., Дюдяева С.Л. Применение технологии дистанционного обучения на примере сельской школы	20
Васильева Н.Ю. Система мультимедийного сопровождения на уроках географии как важнейшее условие успешной социализации обучающихся с нарушением интеллекта	23
Газизова Л.Р. Использование платформы ACADEMIC TESTS для организации и проведения образовательного онлайн-тестирования в процессе дистанционного обучения	27
Гуськова А.Г. Большая перемена: опыт участия школьников в дистанционной проектной деятельности	29
Дронова Р.Е. Онлайн урок как форма реализации дистанционного обучения	33
Зиновьева Ю.А. Формирование познавательного интереса учащихся инженерно – технологического профиля к предмету информатика в условиях дистанционного обучения	37
Умярова Р.А., Ахметова Г.А., Вальшина Г.Р. Роль интерактивной доски на уроках математики при дистанционном обучении	39
Халимова Г.Н. Применение технологий виртуальной и дополненной реальности в образовании	40
Мукина О.Г. О дистанционном обучении в системе повышения квалификации	45
Чипчина Е.Е. Развитие навыков программирования учащихся с использованием онлайн- курсов	49

Сибирев И.В.	
Особенности использования системы контроля версий git при изучении программирования учащимися средних специальных учебных заведений	52
Технологии, методы и методики преподавания	
Витязева О. В.	56
Профессионализация студентов на начальном этапе высшего образования	
Наумова Л.А.	59
Корректировка рабочей программы курса химия в соответствии с «ФГОС 3++»	
Иванова И.С.	62
Использование игровой технологии по химии с учетом принципа адаптивности	
Ахметов М. А.	65
О роли теоретических знаний в подготовке к ЕГЭ по химии	
Литвинова Т.Н., Струков А.Ю.	70
Методическое обеспечение процесса обучения на факультете довузовской подготовки медицинского вуза	
Юркина М.Н.	74
О некоторых методических особенностях преподавания математических дисциплин в университете	
Попов Н.И., Губарь Л.Н.	78
О реализации межпредметных связей курса теории вероятностей и математической статистики с использованием рабочих программ дисциплин	
Воронина И.Н.	81
Формирование территории творчества, как одно из средств активных методов обучения решению математических задач в профильных классах	
Ананичева С.В.	85
Эксперимент и граф в учебном процессе по физике	
Вязовкина О.Н.	87
Методика работы с обучающимися, имеющими риски учебной неуспешности	
Лукьянова И.В., Устюжанина К.В.	91
Координатный метод в курсе геометрии 11 класса	
Волотова И. А., Сибирева А.Р.	94
Алгоритмизация решения текстовых задач на наибольшее и наименьшее значение	
Карапетян А.А., Сибирева А.Р.	98
Обобщения теоремы пифагора – учащимся	
Вихирева С.В.	102
STEAM-технологии – инновации в образовании	

Сибирев И.В.,
преподаватель колледжа информатики и программирования ФГОБУ ВО
«Финансовый университет при Правительстве Российской Федерации»,
(г.Москва)
ivan.sibirev@yandex.ru

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ КОНТРОЛЯ ВЕРСИЙ GIT ПРИ ИЗУЧЕНИИ ПРОГРАММИРОВАНИЯ УЧАЩИМИСЯ СРЕДНИХ СПЕЦИАЛЬНЫХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Системы контроля версий повсеместно используются при разработке программного обеспечения, поэтому актуально их использование в учебном процессе СПО для учащихся, профессии которых будут связаны с программированием.

История систем контроля версий напрямую связана с разработкой операционной системы Linux. CVS – 1980 г., BitKeeper – 2002 г., Subversion – 2004 г., Git – 2005 г.,... (подробнее см.[1]). С тех пор система контроля версий Git плотно вошла в повседневную жизнь разработчика программного обеспечения.

Git позволяет создать центральный репозиторий, клонировать персональный репозиторий (в котором ведется основная работа). Позволяет утягивать в персональный репозиторий ветви изменений программного кода из центрального репозитория, создавать ветви, сливать ветви изменений в одну и отправлять их обратно в центральный репозиторий. Изменения фиксируются в виде последовательности коммитов из проиндексированных изменений файлов.

Основная работа с Git ведется через консоль путем набора и запуска команд Git-интерпретатора. Некоторые учащиеся, обладающие опытом программирования, говорят, что в консоли Git нет ничего сложного. Такого эффекта можно достичь путем прохождения курса «Операционные системы», из которого нас интересуют консоль под Windows CMD/DOS/*.BAT и под Linux *.sh. Рекомендуем также прохождение он-лайн JavaScript тренажера [3] и изучение работы «Pro Git» [2], что сильно расширяет возможности пользователя. Это долгий путь, который в программистской среде считается верным. Проблема в том, что у большинства учащихся недостаточно мотивации и навыков самоорганизации для такой кропотливой работы. А с системой контроля версий Git уже нужно работать.

На подобный случай существуют графические пользовательские интерфейсы (Graphical User Interface, GUI). Нами в лабораторных работах используются «Git 2.8 portable» [3] и «GitExtensions v3.4.1 portable».

Преимущество такого сочетания: свободно распространяемое программное обеспечение, исходный код которого можно посмотреть на Github; программное обеспечение не требует установки и прав администратора, что очень важно в учебных аудиториях. По возможности, программное обеспечение рекомендуется установить – будет быстрее работать. При первом

запуске GitExtensions просит выбрать язык, ввести имя и электронную почту, это нужно для добавления в совершаемые коммиты (фиксации изменений). При желании, по различиям в никнейме можно узнать – с какого компьютера были добавлены изменения.

Минусы GitExtensions. ПО свободно распространяемое, его обновления не всегда успевают за API интерфейсами служб и приложений. Для удаления настроек требуется удалить временные файлы приложения. Доступ к ним можно получить, создав ярлык с адресом «%appdata%», это глобальная переменная Windows, хранящая путь к папке. Еще одна проблема – требуется открытый порт 443 для работы с GitHub. Эту проблему можно преодолеть, воспользовавшись другим GUI GitKraken [6]. Интерфейс уже англоязычный, что вызывает трудности у ряда учащихся. Временные файлы – также по адресу «%appdata%». Требуется связать ПО с Github два раза подряд через браузер. Со второго раза получаем сообщение об успехе.

При работе с Git рекомендуется поднять Git-сервер, чтобы получить возможность доступа к репозиторию из любого персонального компьютера с доступом в Интернет. Для этого можно хранить Git-репозиторий, например, на Яндекс-диске. Вариант – не самый удачный, так как требует установки Яндекс-диска, больше подходит для офисных сотрудников. Другой вариант, «bittorrent-sync» – приложение для синхронизации больших папок через Интернет. Предпочтительно создать в локальной сети общую сетевую папку и завести на ней центральный репозиторий. Эти примеры экзотические, они не самые удачные с точки зрения защиты информации. Git-репозиторий не обладает собственными средствами защиты от несанкционированного доступа, и выкладывать его подобным способом – значит «развешивать приглашения» для юных хакеров.

На предприятии обычно поднимают отдельный GitLab сервер или любой другой сервер с сетевым доступом и защитными оболочками. Теоретически возможно поднять центральный репозиторий на Android телефоне и получать к нему доступ при выходе в Интернет (статьи, повествующие о подобных способах, встречаются в Интернет).

Любители свободного программного обеспечения на практике предпочитают использовать Интернет-сервисы для размещения центральных репозиториях, например, GitHub.

Задание на первую лабораторную работу. Зарегистрироваться на GitHub. Подтвердить регистрацию. Перейти в список «Ваши репозитории». Нажать на кнопку New. Затем перейти в графический пользовательский интерфейс. Нажать на кнопку связи с GitHub. Выбрать репозиторий, папку для размещения на локальном компьютере. Подтвердить клонирование. Ввести логин, пароль (по необходимости). Открыть папку репозитория в проводнике. Создать «*.txt» с любым текстом. Перейти в GUI Git. Нажать на кнопку «коммит» или на кнопку «посмотреть изменения». Добавить изменения файлов в индекс, создать фиксацию или коммит. Нажать на кнопку Push, отправить созданную ветку в центральный репозиторий. Проверить изменения в репозитории на сайте

GitHub. После этого написать об этом отчет со скриншотами. Точно также выложить его на GitHub.

Первая лабораторная работа приводит учащихся к мысли о том, что нужно аккуратно записывать и хранить пароли, работать только в папке персонального репозитория, чтобы иметь возможность централизованно сохранять все изменения.

Чтобы зачистить за собой логины и пароли на учебном компьютере нужно через браузер выйти из аккаунта с сайта GitHub из всех браузеров, через которые заходили; удалить папку репозитория и папку временных файлов (из %appdata%) с жёсткого диска компьютера. Были случаи, когда учащиеся заходили на GitHub под чужим логином и удаляли репозиторий с лабораторными работами сокурсника.

Действия этой лабораторной работы учащиеся повторяют каждый раз в начале пары, работая только в репозитории и «вынимая» из Интернет свои предшествующие разработки. Сначала это идет медленно. Позже доходит до автоматизма или до использования на занятиях личного ноутбука (оба варианта не возбраняются).

Кроме предложенных GitExtensions и GitKraken существуют и другие Git-GUI. OctoGit – родное для GitHub Git-GUI; SmartGit; Team Explorer – плагин для Visual Studio 2019 (для работы с GitHub дополнительного расширения под названием «GitExtensions»); и тд. В Интернет встречаются самописные плагины под различные текстовые редакторы и *.bat скрипты. Будьте внимательны и обходите их стороной, если не удастся разобраться в их исходном коде. Используйте только широко известные и распространенные Git-GUI. Может оказаться, что самописные и небрежно скаченные Git-GUI «подворовывают» логины и пароли от GitHub или привносят сетевую уязвимость в локальную сеть. Поэтому на предприятии стараются пользоваться официальным программным обеспечением или консолью Git. Тем не менее, самописные Git-GUI являются неплохим развлечением для тренирующихся или скучающих программистов.

В последнее время GitHub используется как портфолио проектов программиста при трудоустройстве. В качестве таких проектов начинающего программиста иногда могут послужить аккуратно сложенные лабораторные работы, к которым, кроме того, можно возвращаться позднее в справочных целях.

Если учащийся обладает определенным навыком общения с GitHub, то ответом на заданный им вопрос может быть ссылка на проект преподавателя, при условии что он хорошо прокомментирован. Само собой – это только способ начать диалог!

Git/GitHub изначально использовался как средство командной разработки, потому его использование в курсовых и дипломных проектах является неоспоримым плюсом. Но знакомиться с подобными технологиями лучше задолго до того, как они потребуются всерьез, а именно на лабораторных и практических работах по информатике или по программированию.

ЛИТЕРАТУРА

1. Url:<https://techrocks.ru/2019/02/19/git-origin-story/>
2. Pro Git Scott Chacon, Ben Straub Version 2.1.266, 2020-10-03
Url:<https://git-scm.com/book/en/v2>
3. Учим Git. Url:<https://learngitbranching.js.org/>
4. Dist_Url_Git. Url:<https://github.com/git-for-windows/git/releases/download/v2.28.0.windows.1/PortableGit-2.28.0-32-bit.7z.exe>
5. Github.com, GitExtensions.Url: <https://github.com/gitextensions/gitextensions/releases/tag/v3.4.1>
6. GitKraken. Url:<https://www.gitkraken.com/>