

УДК 37.091.212:004.773.7  
DOI: 10.24144/2524-0609.2021.49.107-110

**Мосіюк Олександр Олександрович**  
кандидат педагогічних наук  
доцент кафедри комп'ютерних наук та інформаційних технологій  
Житомирський державний університет імені Івана Франка  
м. Житомир, Україна  
mosxandrwork@gmail.com  
<http://orcid.org/0000-0003-3530-1359>

## ОСОБЛИВОСТІ НАВЧАННЯ МАЙБУТНІХ УЧИТЕЛІВ ІНФОРМАТИКИ РОБОТІ ІЗ СИСТЕМОЮ КОНТРОЛЮ ВЕРСІЙ GIT

**Анотація.** Робота у команді вважається чи не найважливішою навичкою сучасного фахівця, а тим більше педагога. Безперечним є той факт, що починати розвивати її необхідно ще у школі, зокрема і на уроках інформатики. Тому майбутній вчитель інформатики має знати СКВ та застосовувати їх для організації спільної роботи учнів як на заняттях так і у процесі підготовки учнівських робіт для МАН. У той же час ця тематика не є достатньо розробленою, а отже вибрана тема статті є актуальною, а її мета полягає в розкритті основних питань навчання роботи студентів із системами контролю версій. Під час дослідження використовувалися такі методи як: аналіз науково-методичних джерел; опис і порівняння результатів; узагальнення. Загалом у статті аналізується особливості навчання майбутніх учителів інформатики роботі з Git і наголошується на важливості засвоєння педагогами відповідного програмного забезпечення. Звертається увага на важливих моментах вивчення команд Git та терміналу Linux для коментування; зауважується на можливостях застосування такого онлайн сервісу як GitHub Classroom.

**Ключові слова:** підготовка майбутніх учителів інформатики; система контролю версіями; Git; ресурс GitHub.com; GitHub Classroom.

**Вступ.** Навчання багатьох предметів, пов'язаних із програмування починається із засвоєнням основних синтаксичних конструкцій мови та їх призначення. Наприклад, обов'язково розглядаються типи даних, конструкції умовного оператора, циклів, функцій, концепції функціонального та об'єктно-орієнтованого програмування тощо. При вивченні верстки web-сторінок докладно описують поняття мови гіпертекстової розмітки, каскадних таблиць стилів та JavaScript, а також розкривають стандарти верстки, порівнюються можливості технологій необхідних для створення сайтів, визначаються їх переваги та недоліки. Як наслідок, основна частина науково-методичних праць пов'язана саме із цими питаннями.

В той же час, питанням командної роботи групи студентів або ж учнів при виконанні лабораторних чи практичних робіт із використанням систем контролю версій увага приділяється значно менше. На підтвердження цієї тези, наведемо короткий аналіз наявних науково-педагогічних джерел.

**Аналіз останніх досліджень та публікацій.** При аналізі наукової літератури, що стосується цього питання, варто звернути на ряд особливостей. Перша з них, яку варто виокремити, – це зосередженість на вивченні алгоритмів програмування. Так загальною теорією методики навчання алгоритмізації викладено у працях Н.В.Морзе [9]. А.М.Стрюк описує методичні аспекти навчання системному програмуванню майбутніх інженерів-програмістів [10]. Проблематику вивчення об'єктно-орієнтованого підходу до програмування розглядав С.А.Вернигоренко [3].

Окремі аспекти пов'язані із методикою навчання програмування студентів різних педагогічних та технічних спеціальностей, а також учнів шкіл розкривали Ю.П.Біляй [2], П.Г.Шевчук [12], О.Б.Ящик [13] та інші.

Окремо варто назвати фахівців, які досліджували тематику пов'язану із вивченням верстки та програмування web-сайтів: В.М.Базурін [1], А.В.Ворожбит, Н.В.Дегтярьова [6], О.С.Рибак [4].

Звичайно ці теми є важливими для формування фундаментальних знань із предметів, пов'язаних із

алгоритмами, розробкою програмного забезпечення та web-сайтів, проте вони не враховують такий важливий компонент як робота у команді, формування культури оформлення коду та, власне, контроль за версіями у процесі розробки. Цей компонент підготовки є не менш значимим для сучасного фахівця. У цьому контексті розуміння роботи технологій систем контролю версій є вельми важливою професійною складовою для майбутніх вчителів і викладачів інформатики, інформаційних технологій та програмування.

Проте необхідно зазначити, що питання, пов'язані із набуттям студентами компетенцій, які дозволяють забезпечити реєстрацію змін у програмному коді і синхронізувати процес написання, власне, коду програми в команді фахівців, є недостатньо висвітлених у науково-методичній літературі. Так О.В.Циба тезово описує використання Git у рамках дистанційного навчання програмуванню [11]. Використання розподілених систем контролю для організації командної роботи над проектом докладно розкривають В.Г.Гриценко і О.М.Подольян [5], зокрема у статті окреслено процедуру та ключові кроки розробки навчального проекту і продемонстровано практичну сторону роботи в команді розробників.

Звичайно, що цього не достатньо, для повноцінного висвітлення тематики використання розподілених систем контролю версій у освітньому процесі, тому мета цієї статті полягає розкритті основних питань навчання роботи із систем контролю версій на прикладі програми Git у процесі підготовки майбутніх учителів інформатики.

**Методи дослідження.** Методологічною основою дослідження став системний підхід, завдяки якому було узагальнено напрацювання як українських так і закордонних фахівців, що докладно вивчали методичні особливості навчання програмування у школі та вищих закладах освіти. У рамках зазначеного підходу використовувалися такі теоретичні загальнонаукові методи як: аналіз наукових та методичних джерел провідних фахових видань України; опис і порівняння результатів, які були у них представлені; узагальнення досвіду провідних фахівців;

синтез тощо.

**Виклад основного матеріалу.** Відповідно до Закону України «Про освіту»: «... метою освіти є всебічний розвиток людини як особистості та найвищої цінності суспільства, її талантів, інтелектуальних, творчих і фізичних здібностей, формування цінностей і необхідних для успішної самореалізації компетентностей ... » [7], а отже важливою є якісна підготовка педагогів, здатних забезпечити навчальний та виховний процес в сучасних умовах. А у контексті природничо-математичних наук важливим є «... формування вмінь практичного і творчого застосування здобутих знань» у прикладній сфері, набуття необхідних навичок комунікації та роботи у команді фахівців [8]. Тому для майбутніх учителів інформатики надзвичайно важливим є володіння технологіями, які забезпечують професійне керування процесом створення програмного забезпечення. Одними з найважливіших із них є системи керування версіями (Subversion, Team Foundation Server, Git, Mercurial, Bazaar), які можуть бути як централізованими так і розподіленими. Серед них, найбільш поширеною та доступною є Git.

Зауважимо, що система керування версіями Git має значні можливості для використання її у навчальному процесі підготовки майбутніх учителів інформатики. Отже спершу розглянемо докладніше сутність спеціалізованого програмного забезпечення.

Отже, *системами контролю версій* (також буде використовувати скорочення СКВ) називаються спеціалізовані програми, які дозволяють реєструвати зміни в одному або декількох файлах проекту, щоб у подальшому була можливість повернутися до попередніх етапів розробки, у разі виникнення певних критичних ситуацій [5, с.251]. У свою чергу, Git – це розподілена система контролю версій, яка дозволяє вести одночасну розробку програмного забезпечення команді програмістів і при цьому взаємодіяти між собою та віддаленими репозиторіями [14].

Програма Git існує у декількох версіях: із графічним користувацьким інтерфейсом і на базі командного рядка. Проте розглядатимемо основні питання навчання технології саме для другого випадку. Такий підхід дозволить майбутнім учителям інформатики зрозуміти всі особливості роботи із СКВ та, у подальшому, пояснити її учням.

Першим кроком варто роз'яснити особливості встановлення СКВ на комп'ютер. Для персонального комп'ютера із ОС Linux встановлення відповідної програми є чи найпростішим. Так, якщо на ПК встановлено дистрибутив Fedora або інший, який базується на його основі, то у терміналі слід ввести команду `sudo yum install git`. У випадку використання операційних систем, для яких основою є Debian, команда встановлення матиме вигляд такий – `sudo apt-get install git`. Після введення відповідних інструкцій система буде розгорнута та будуть налаштовані всі необхідні взаємозв'язки.

Дещо більше операцій необхідно виконати для розгортання цієї системи на ОС Windows. Щоб почати використовувати Git на ПК необхідно вибрати один із запропонованих варіантів: скористатися програмою із графічним інтерфейсом; встановити утиліту gitbash, яка імітує роботу терміналу Linux, його основні команди і, власне, самі команди git; використати емулятор командного рядка Cmder, а саме його розширений варіант.

СКВ Git повністю повторює роботу із файлами у терміналі, оскільки вона створювалася для операційної системи Linux. Тож важливим моментом при вивченні програми є повторення основних команд

терміналу Linux.

Тепер, власне, розглянемо яким чином необхідно використовувати Git. Для початку варто наголосити студентам, що будь-яка їхня робота (лабораторна чи практична) із програмування або верстки є певним проектом, який слід виконати. При цьому доцільно дотримуватися наступних правил при організації робочого простору:

- для кожного проекту необхідно виділяти окрему папку, де розмішуватимуться всі робочі файли (такий каталог можна створити як за допомогою графічного інструментарію операційної системи, так і у командному рядку);

- важливі зміни, які були внесені у проект мають одразу ж коментуватися із повним описом;

- експериментальні підходи до вирішення певної проблемної ситуації мають виділятися в окремі гілки розвитку проекту і об'єднуватися із основною, якщо новий підхід мав успіх;

- обов'язковим є синхронізація із хмарним сховищем (GitHub або GitLab), що дозволить команді із учнів або студентів спільно працювати над проектом.

Подальше вивчення програми варто розділити на три основні частини: створення локального репозиторію, додавання нових файлів та коментування змін; навігація по репозиторію і створення нових гілок розвитку проекту, їх об'єднання; синхронізація локального та глобального репозиторіїв (розміщених у таких хмарних сховищах як GitHub або ж GitLab).

Пояснення першої частини варто почати із прикладу виконання глобальних налаштувань Git та ініціалізації проекту, щоб СКВ почала відслідковувати всі зміни, які вносяться у файли. Для цього треба перейти у директорію проекту і виконати інструкцію `git init`. У результаті виконання команди, система контролю версій створить піддиректорію `.git` у вказаній папці, яка міститиме всі необхідні файли для коректної роботи СКВ із цим репозиторієм.

Тепер пояснимо сутність останньої дії. При роботі над дипломним проектом, виконанні лабораторного чи практичного завдання у студента виникатимуть різні ідеї, які необхідно реалізувати і, при цьому, контролювати, а, у деяких випадках, повертатися до попередніх версій проекту. Все це необхідно зберігати та систематизувати. Фактично студент, користуючись Git-ом, зберігає певний стан проекту у визначеній точці і, у разі необхідності, завжди може повернутися на декілька кроків назад і не втратити напрацювань.

На цьому ж прикладі також варто продемонструвати команди для: перевірки статусу файлів (`git status`), перегляду змін у файлах (`git diff`) підготовки файлів до коментування (`git add`), власне, виконання коментування (`git commit`) тощо. Важливо зауважити, що файли, які мають ігноруватися системою контролю версій мають вноситися у файл `.gitignore`, що також варто пояснювати студентам.

Роз'яснення навігації варто розпочати із команди, яка демонструє історію створення «комітів» (`git log`). Тут необхідно зауважити на таких моментах як: біля ключового слова `commit` завжди виводиться унікальне хеш значення, яке створюється при коментуванні і є унікальним та однозначно ідентифікує зафіксований СКВ стан студентського проекту; у дужках запис `HEAD -> master` вказує, що існує основна лінія (гілка у термінах Git) розробки, де кожен наступний коміт успадковує свій стан від попереднього, лише із незначними змінами, а також вказується автор та дата коментування.

Також необхідно продемонструвати яким чином можна: переглянути всі доступні гілки, створити

нову, змінити назву, видаляти їх (*git branch*); перейти на вказаний «коміт» або змінити гілку розробки (*git checkout*); об'єднати гілки розробки і представити це як один «коміт» (*git marge*).

Спільна робота над студентським проектом не можлива без використання хмарних репозиторіїв. Найбільш відомими з них є GitHub та GitLab. Найоптимальніший вибір для студентських проектів є сервіс GitHub і тому, саме на його прикладі варто продемонструвати студентам створення віддаленого репозиторію, поєднання його із локальним (*git remote*), завантаження всіх змін, які були внесені розробниками (*git fetch, git pull*), вивантаження «комітів» із локального репозиторію у глобальний (*git push*).

Важливою перевагою використання сервісу GitHub є можливість створити спільне робоче середовище для учнів та викладачів, які викладають інформатику та програмування за допомогою GitHub Classroom. Це дозволяє не тільки навчити студентів працювати із системою керування версіями Git, а й створювати спеціалізовані завдання для програму-

вання, виконувати їх перевірку, спільно реалізувати навчальні проекти.

**Висновки та перспективи подальших досліджень.** Підводячи підсумки зауважимо на ряд важливих моментах: знання систем керування версіями (зокрема Git) є важливою частиною підготовки майбутніх учителів інформатики; навчання студентів роботі із такими програмами необхідно поєднувати із вивченням програмування та заохочувати виконувати лабораторні і практичні завдання із використанням Git; практична складова є чи найважливішою частиною опанування СКВ студентами, а отже викладачі мають демонструвати на прикладах правильну організацію та культуру роботи із такими програмами.

Серед подальших досліджень варто виділити такі перспективні напрями: докладне вивчення організації групової проектної роботи студентів у рамках виконання курсових і дипломних робіт; організація навчального онлайн процесу із програмування та web-розробки за допомогою GitHub Classroom; перспективи використання системи Git в навчанні програмування школярів.

### Список використаної літератури

1. Базурін В. М. Особливості навчання веб-програмування мовою JavaScript студентів-математиків. Вісник Житомирського державного університету, 2014. Вип. 1 (73). С.79–83.
2. Білій Ю. П. Реактивне програмування. Науковий часопис НПУ імені М.П.Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання, 2018. № 20 (27). С.124–130.
3. Вернигоренко С. А. Вивчення основ об'єктно орієнтованого програмування у класах фізико-математичного профілю. Науковий часопис НПУ імені М.П.Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання. 2007. № 5 (12). С.63–74.
4. Ворожбит А.В., Рибак О.С. Огляд курсу за вибором «Основи верстки та веб-програмування». Фізико-математична освіта. 2018. Вип. 1 (15). С.20–27.
5. Грищенко В. Г., Подолян О. М. Використання системи управління версіями GIT для організації командної роботи над IT проектом. Інформаційні технології і засоби навчання. 2014. Том 39, №1. С.250–263.
6. Дегтярьова Н. В. Методичні особливості навчання студентів розмітки блоків при вивченні таблиць каскадних стилів. Фізико-математична освіта. 2017. Випуск1 (11). С.32–36.
7. Закон України «Про освіту». URL: <https://zakon.rada.gov.ua/laws/show/2145-19#top> (дата звернення: 12.04.2021).
8. Концепція розвитку природничо-математичної освіти (STEM-освіти). URL: <https://zakon.rada.gov.ua/laws/show/960-2020-%D1%80#top> (дата звернення: 12.04.2021).
9. Морзе Н.В. *Методика навчання інформатики* / за ред. М.І.Жалдака. У 4 ч. Київ: Навчальна книга, 2004. Ч. 4. Методика навчання основ алгоритмізації та програмування. 368 с.
10. Стрюк А.М. Системне програмування у підготовці фахівців з програмної інженерії. Збірник наукових праць Уманського державного педагогічного університету. 2011. Ч. 3. С.260–271.
11. Цыба О.В. *Использование системы контроля версий Git в дистанционном обучении программированию*. URL: <http://dspace.puet.edu.ua/handle/123456789/5589> (дата звернення: 12.04.2021).
12. Шевчук П.Г. Програмно-технологічні умови використання мови C# для навчання програмування в загальноосвітніх навчальних закладах. Науковий часопис НПУ імені М.П. Драгоманова. Серія 2. Комп'ютерно-орієнтовані системи навчання. 2010. № 10 (17). С.80–84.
13. Ящик О.Б. Машина Тюрінга як універсальний виконавець алгоритмів та її застосування в процесі поглибленого вивчення алгоритмізації і основ програмування старшокласниками. Інформаційні технології і засоби навчання. 2016. Том 52, №2. С.10–17.
14. A short history of Git. URL: <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git> (дата звернення: 12.04.2021).

### References

1. Bazurin, V. M. (2014). Osoblyvosti navchannia veb-prohramuvannia movoiu JavaScript studentiv-matematykiv [Features of learning web programming in JavaScript for math students]. *Visnyk Zhytomyrskoho derzhavnoho universytetu*, 1 (73), 79–83. [in Ukrainian].
2. Bilii, Yu. P. (2018). Reaktyvne prohramuvannia [Reactive programming]. *Naukovyi chasopys NPU imeni M.P.Drahomanova. Serii 2. Kompiuterno-orientovani systemy navchannia*, 20 (27), 124–130. [in Ukrainian].
3. Vernyhorenko, S. A. (2007). Vyvchennia osnov obiektno oriientovanoho prohramuvannia u klasakh fizyko-matematychnoho profiliiu [Learning the basics of object-oriented programming in physics and math lessons]. *Naukovyi chasopys NPU imeni M. P. Drahomanova. Serii 2. Kompiuterno-oriientovani systemy navchannia*, 5 (12), 63–74. [in Ukrainian].
4. Vorozhbit, A. V., & Rybak, O. S. (2018). Ohliad kursu za vyborom «Osnovy verstky ta veb-prohramuvannia» [Overview of elective courses «Basics of layout and web programming»]. *Fizyko-matematychna osvita*, 1 (15), 20–27. [in Ukrainian].
5. Hrytsenko, V. H., & Podolian, O. M. (2014). Vykorystannia systemy upravlinnia versiiamy GIT dlia orhanizatsii komandnoi roboty nad IT proektom [Using the GIT version control system to organize teamwork on an IT project]. *Informatsiini tekhnolohii i zasoby navchannia*, 1 (39), 250–263. [in Ukrainian].
6. Dehtiarova, N. V. (2017). Metodychni osoblyvosti navchannia studentiv rozmitky bloktiv pry vyvchenni tablyts kaskadnykh styliv [Methodical features of teaching students to mark blocks in the study of cascading style sheets]. *Fizyko-matematychna osvita*, 1 (11), 32–36. [in Ukrainian].
7. *Zakon Ukrainy «Pro osvitu» [Law of Ukraine «On Education»]*. <https://zakon.rada.gov.ua/laws/show/2145-19#top>. [in Ukrainian].
8. Kontseptsiiia rozvytku pryrodnycho-matematychnoi osvity (STEM-osvity) [The concept of development of natural and mathematical education (STEM-education)]. <https://zakon.rada.gov.ua/laws/show/960-2020-%D1%80#top> [in Ukrainian].
9. Morze, N. V. (2004). *Metodyka navchannia informatyky. Ch. 4. Metodyka navchannia osnov alhorytmizatsii ta prohramuvannia*.

- [Methods of teaching computer science. Part 4. Teaching methodology the basics of algorithms and programming ]. Navchalna knyha. [in Ukrainian].
10. Striuk, A. M. (2011). Systemne prohramuvannia u pidhotovtsi fakhivtsiv z prohramnoi inzhenerii [System programming in the training of specialists in software engineering]. *Zbirnyk naukovykh prats Umanskoho derzhavnoho pedahohichnoho universytetu*, 3, 260–271. [in Ukrainian].
  11. Tsyba, O. V. *Ispolzovanie sistemy kontrolya versiy Git v distantsionnom obuchenii programmirovaniyu* [Using the Git version control system in distance learning programming]. <http://dspace.puet.edu.ua/handle/123456789/5589> [in Ukrainian].
  12. Shevchuk, P. H. (2010). Prohramno-tekhnolohichni umovy vykorystannia movy C# dlia navchannia prohramuvannia v zahalnoosvitnikh navchalnykh zakladakh [Software and technological conditions for the use of the C # language for teaching programming in secondary schools.]. *Naukovyi chasopys NPU imeni M. P. Drahomanova. Serii 2. Kompiuterno-oriientovani systemy navchannia*, 10 (17), 80–84. [in Ukrainian].
  13. Yashchuk, O. B. (2016). Mashyna Tiurinha yak universalnyi vykonavets alhorytmiv ta yii zastosuvannia v protsesi pohlyblyenoho vyvchennia alhorytmizatsii i osnov prohramuvannia starshoklasnykamy [Turing machine as a universal executor of algorithms and its application in the process of in-depth study of algorithmization and basics of programming by high school students]. *Informatsiini tekhnolohii i zasoby navchannia*, 2 (52), 10–17. [in Ukrainian].
  14. A short history of Git. <https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git>.

Стаття надійшла до редакції 13.07.2021 р.

Стаття прийнята до друку 18.07.2021 р.

**Mosiuk Oleksandr**

Candidate of Pedagogical Sciences, Ph.D., Senior Lecturer  
Department of Computer Science and Information Technology  
Zhytomyr Ivan Franko State University, Zhytomyr, Ukraine

### FEATURES OF TRAINING OF FUTURE INFORMATICS TEACHERS WORKING WITH THE GIT VERSION CONTROL SYSTEM GIT

**Abstract.** Teamwork is the most important soft skill of the modern specialist, not to mention a teacher. It is also undisputed that skills development is very important to start at school, in particular at computer science lessons. Accordingly, a future teacher of computer science has to know version control systems and use one of them to organize collective student's work like in class and in the process of preparing for science competitions of the Junior Academy of Sciences of Ukraine. This topic of the article is relevant because it doesn't fully describe in the scientific and methodological resources. The main goal of this article is to open the main features of teaching students to work with version control systems on the example of Git. The systematic approach was the methodological basis for the investigation discovery. But besides, the researcher analyzed, compared, and summarized the investigation and experience of the leading scientists. As the result of the article the author rise to the following conclusions that the knowledge of version control systems (in particular Git) is an important part of the preparation of computer science teachers; teaching students to work with such programs have to be combined with the study of programming; should be encouraged students to complete labs and case studies using Git; teachers have to demonstrate through examples the correct organization and culture of work with such programs. The writer is also considering using the GitHub Classroom service to organize the educational process subjects such as programming and page making of web pages.

**Key words:** training of future computer science teachers; version control system; Git; resource GitHub.com; GitHub Classroom.