МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра 105 «Інформаційних технологій проектування»

ЗАТВЕРДЖУЮ

Голова НМК 2

_____ Дмитро КРИЦЬКИЙ

«_____» _____ 2024 року

РОБОЧА ПРОГРАМА
ВИБІРКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ
(назва навчальної дисципліни)

Галузь знань: 10 «Природничі науки»,
11 «Математика та статистика»,
12 «Інформаційні технології» ,
15 «Автоматизація та приладобудування»,
16 «Хімічна та біоінженерія» ,
17 «Електроніка та телекомунікації»,
19 «Архітектура та будівництво»
(шифр і назва галузі знань)

Спеціальність: для усіх спеціальностей зазначених галузей
(шифр і назва спеціальностей)

Освітня програма: для усіх освітніх програм зазначених галузей
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: другий (магістерський)

Харків 2024 рік

Розробник: Каратанов О. В., доцент кафедри 105 «Інформаційних
(прізвище та ініціали, посада, наукова ступінь та вчене звання)
технологій проектування», к.т.н., доцент _____
(підпис)

Робочу програму розглянуто на засіданні кафедри інформаційних технологій проектування
(назва кафедри)

Протокол № 1 від 31.08.2024 року

В.о. завідувача кафедри 105 А.М. БИКОВ
(наукова ступінь та вчене звання) (підпис) (ініціали та прізвище)

1 Опис навчальної дисципліни

<table>
<tr><td rowspan="2">Найменування показників</td><td rowspan="2">Галузь знань, спеціальність, освітня програма, рівень вищої освіти</td><td>Характеристика навчальної дисципліни</td></tr>
<tr><td>денна форма навчання</td></tr>
<tr><td>Кількість кредитів – 5</td><td rowspan="13">Галузь знань
12 «Інформаційні технології»,
10 «Природничі науки»,
11 «Математика та статистика»,
12 «Інформаційні технології»,
15 «Автоматизація та приладобудування», 16 «Хімічна та біоінженерія», 17 «Електроніка та телекомунікації»,
19 «Архітектура та будівництво»
Рівень вищої освіти:
другий (магістерський)</td><td>Вибіркова</td></tr>
<tr><td>Модулів – 2</td><td>Навчальний рік</td></tr>
<tr><td>Змістових модулів – 4</td><td>2024/2025</td></tr>
<tr><td rowspan="2">Загальна кількість годин – 150</td><td>Семестр</td></tr>
<tr><td>1-й</td></tr>
<tr><td rowspan="6">Тижневих годин для денної форми навчання:
аудиторних – 3
самостійної роботи студента – 5,6</td><td>Лекції*</td></tr>
<tr><td>32 год.</td></tr>
<tr><td>Практичні, семінарські</td></tr>
<tr><td>0 год.</td></tr>
<tr><td>Лабораторні*</td></tr>
<tr><td>32 год.</td></tr>
<tr><td>Самостійна робота</td></tr>
<tr><td>86 год.</td></tr>
<tr><td>Вид контролю:</td></tr>
<tr><td>іспит</td></tr>
</table>

Співвідношення кількості годин ауд. занять до сам. роботи становить для денної форми 64 / 86 = 0,74.

*Аудиторне навантаження може бути зменшеним або збільшеним на одну годину в залежності від розкладу занять.

2 Мета та завдання навчальної дисципліни

Об’єкт вивчення – процеси функціонування систем контролю версій та створення програмного забезпечення при їх використанні.

Предмет вивчення – системи контролю версій.

Мета навчання – надання слухачам знань, уміння, навичок та методичних прийомів, що необхідні для використання систем контролю версій.

Завданнями дисципліни є:

– опанувати базові знання щодо основ контролю версій за допомогою Git;

– освоїти сучасні підходи до розподіленої розробки;

– отримати знання з сучасних технологій контролю версій;

– ознайомитись з архітектурою розподілених систем контролю версій;

– вивчити основні команди Git та опанувати графічні клієнти Git.

Студенти повинні досягти таких компетентностей:

– здатність до абстрактного мислення, аналізу та синтезу;

– здатність застосовувати знання у практичних ситуаціях;

– навички використання інформаційних і комунікаційних технологій;

– навички роботи з інформацією (уміння знаходити та аналізувати інформацію з різних джерел);

– здатність генерувати нові ідеї (креативність);

– здатність приймати обґрунтовані рішення;

– здатність оцінювати та забезпечувати якість виконуваних робіт;

– визначеність і наполегливість щодо поставлених завдань і взятих обов’язків.

В результаті засвоєння курсу студенти повинні

знати:

– область використання систем контролю версій;

– класифікацію систем контролю версій;

– типові підходи при роботі з такими системами;

– правила та методи роботи та використання систем контролю версій при реалізації ІТ-проектів.

вміти:

– використовувати на практиці теоретичні знання з питань використання систем контролю версій, вибору найбільш ефективних рішень щодо їх використання;
– створювати та використовувати локальні та віддалені репозиторії;
– взаємодіяти у командній розробці програмного забезпечення.

мати навички:

– використання Git.

3 Програма навчальної дисципліни

3.1 Лекції

Змістовний модуль 1. Загальні принципи роботи систем контролю версій

Тема 1. Системи контролю версій

Визначення систем контролю версій (СКВ). Історія розвитку СКВ. Моделі керування версіями: Lock-Modify-Unlock, Copy-Modify-Merge. Можливості СКВ. Класифікація СКВ. Відповідаємо на питання, чому Git став універсальним інструментом, з якого починається практично будь-який проект у розробці. Посилання для самостійного вивчення

Тема 2. Основні команди Git

Встановлення Git. Налаштування операційної системи – *git config* (Ubuntu/MacOS/Windows, примітка – CRLF),. Знайомство з основами Git. Перші команди git. Докладно розбираємо процес від початку роботи до фіксації результату в Git: як створити репозиторій (*git init*), додати файл (git add) і зробити комміт (git commit).

Тема 3. Основні команди Git. Частина 2

Аналіз зроблених змін. Знайомимось із командою git diff, яку потрібно обов'язково запускати перед кожним коммітом. Аналіз історії змін (коммітів). Вчимося отримувати різноманітну інформацію про минулі комміти: хто, коли і як змінював код. Вивчаємо команди, які дозволять вирішити це завдання: log, show, blame, grep. Скасування змін у робочій директорії. Відкат змін, які зроблено в робочій директорії, але ще не потрапили до комміту. Вчимося повертати змінені файли до початкового стану. Скасування коммітів. Що робити, якщо коміт уже зроблено, але з якихось причин нас не влаштовує? Вивчаємо спеціальні команди, що дозволяють спростити скасування або зміну комміту: revert, reset. Зміна останнього комміту. Зробили коміт, але забули додати частину файлів? Розбираємось, як додати зміни до поточного комітету, не створюючи новий.

Тема 4. Гілки

Створення гілок. Створення та перемикання гілок. Переміщення HEAD. Журнал для всіх гілок. Видалення гілок. Перейменування гілок. Перевага гілок у Git. Злиття гілок: Fast-forward merge, тристороннє злиття. Вирішення конфліктів. Перебазування гілок. Cherry-pick.

Тема 5. Віддалені репозиторії

Інтеграція з Github. Створення облікового запису на Github. Вчимося налаштовувати GitHub, створювати в ньому репозиторій та з'єднувати його з локальним репозиторієм. А також клонувати репозиторій, створений на GitHub на свій комп'ютер. GitHub Remote – Push & Pull.

Тема 6 Усередині Git

Stash (як «ховати» зміни у робочій директорії та відновлювати їх за необхідності?). .gitignore (ігнорування файлів: вивчаємо поняття "ігнорування файлів" і вчимося правильно визначати, що додавати до репозиторій, а що ні). Основні частини Git: Робоча директорія (Working Directory), Індекс, Репозиторій. Знімки. Внутрішній устрій git.

Тема 7 Огляд систем контролю версій

Історія СКВ. Рейтинги СКВ. Source Code Control System (SCCS). Interleaved deltas. RCS (Revision Control System). Система керування паралельними версіями (Concurrent Versions System). Subversion. Git. Mercurial.

Змістовний модуль 2. Розподілена розробка програмного забезпечення

Тема 8. Git. Графічні інтерфейси

gitk. git gui. GitHub Desktop. Fork. Tower. Sourcetree. SmartGit. Відкриті проекти (Open Source). Як влитися у розробку відкритих проектів, навчиться командній роботі та покращити своє портфоліо.

Тема 9. Стратегії використання гілок
Centralized Workflow (централізована стратегія). Feature Branch Workflow. Gitflow Workflow. Integration Manager Workflow. Dictator and Lieutenants Workflow. Forking Workflow
Тема 10. Стилі найменування коммітів
Домовленість щодо найменування коммітів. Типи комміту.
Тема 11. Інструменти Git
Вибір ревізій. Інтерактивне індексування. Хования та чищення. Підписання праці. Пошук (git blame, git bisect). Переписування історії. Складне злиття. Rerere.
Тема 12. Віртуальна команда
Розподілена розробка. Що таке віртуальна команда? Як підвищити ефективність роботи віртуальної команди?

4 Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л.	п. р.	лаб.	к. р.	с. р.
Змістовий модуль 1						
Тема 1	4	2	–	–	–	2
Тема 2	24	4	–	8	–	12
Тема 3	22	4	–	8	–	10
Тема 4	24	4	–	8	–	12
Тема 5	20	2	–	8	–	10
Тема 6	4	2	–	–	–	2
Тема 7	4	2	–	–	–	2
Разом за змістовим модулем 1	92	20	–	32	–	50
Змістовий модуль 2						
Тема 8	10	2	–	–	–	8
Тема 9	10	4	–	–	–	6
Тема 10	10	2	–	–	–	8
Тема 11	10	2	–	–	–	8
Тема 12	8	2	–	–	–	6
Разом за змістовим модулем 2	48	12	–	–	–	36
Усього годин за дисципліною	**150**	**32**	**–**	**32**	**–**	**86**

5 Теми семінарських занять
Семінарські заняття навчальним планом не передбачені.

6 Теми практичних занять
Практичні заняття навчальним планом не передбачені.

7 Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Основні команди Git	12
2	Робота з гілками у Git	8
3	Видалені репозиторії в Git	8
4	Модульна контрольна робота	4
	Разом	**32**

8 Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Поточна самостійна робота: робота з літературними джерелами та програмною документацією, робота над конспектом та тестовими завданнями, підготовка до участі в олімпіаді з програмування	86
	Разом	**86**

9 Курсова робота

Курсова робота навчальним планом не передбачена.

10 Методи навчання

Лекції проводяться з використанням основних розділів конспекту лекцій в електронній формі, елементів мультимедійної підтримки курсу (презентації), демонстрацій окремих прийомів роботи з інструментальним середовищем та/або роздаточного матеріалу у вигляді схем та діаграм.

Лабораторні роботи виконуються з використанням системи контролю версій Git.

Самостійна робота включає підготовку до лабораторних робіт, модульного контролю та заліку і вивчення вказаних вище тем за конспектом, літературними джерелами та програмною документацією.

11 Методи контролю

Поточний контроль здійснюється відповідно до повноти, своєчасності та якості виконання лабораторних робіт.

Проміжний (модульний) контроль – тестова контрольна робота на 7 та 14-му тижні.

Підсумковий контроль – письмовий залік та усні відповіді на запитання викладача за курсом.

12 Критерії оцінювання та розподіл балів, які отримують студенти

12.1 Розподіл балів, які отримують студенти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модульні контрольні роботи	0–50	2	0–50
Лабораторні роботи	1–5	8	8–40
Робота на лекціях	0–1	10	0–10
		Усього за семестр	8–100

Семестровий контроль (іспит/залік) проводиться у разі відмови студента від балів поточного тестування та за наявності допуску до іспиту/заліку. При складанні семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

12.2 Якісні критерії оцінювання

Проміжна атестація включає в себе теоретичні завдання, що дозволяють оцінити рівень засвоєння учнями знань, і практичні завдання, що виявляють ступінь сформованості умінь і володінь.

Засвоєні знання і освоєння вміння перевіряються за допомогою модулів, вміння і володіння перевіряються в ході вирішення завдань.

Обсяг і якість освоєння учнями дисципліни, рівень сформованості дисциплінарних компетенцій, оцінюються за результатами поточних і проміжної атестацій кількісною оцінкою, вираженої в балах, максимальна сума балів з дисципліни дорівнює 100 балам.

12.3 Критерії оцінювання роботи студента протягом семестру

Сума балів, набраних студентом з дисципліни, переводиться в оцінку відповідно до таблиці.

Сума балів з дисципліни	Оцінка за проміжної атестації	Характеристика рівня освоєння дисципліни
від 90 до 100	«зараховано» / «відмінно»	Студент демонструє сформованість дисциплінарних компетенції на підсумковому рівні, виявляє всебічне, систематичне і глибоке знання навчального матеріалу, засвоїв основну літературу і знайомий з додатковою літературою, рекомендованою програмою, вміє вільно виконувати практичні завдання, передбачені програмою, вільно оперує набутими знаннями, вміннями, застосовує їх у ситуаціях підвищеної складності.
від 75 до 89	«зараховано» / «добре»	Студент демонструє сформованість дисциплінарних компетенції на середньому рівні: основні знання, вміння освоєні, але допускаються незначні помилки, неточності, труднощі при аналітичних операціях, перенесення знань і умінь на нові, нестандартні ситуації.
від 60 до 74	«зараховано» / «задовільно»	Студент демонструє сформованість дисциплінарних компетенції на базовому рівні: в ході контрольних заходів допускаються значні помилки, виявляється відсутність окремих знань, умінь, навичок за деякими дисциплінарним компетенціями, студент відчуває значні труднощі при оперуванні знаннями та вміннями при їх перенесенні на нові ситуації.
від 41 до 59	«не зараховано» / «незадовільно»	Студент демонструє сформованість дисциплінарних компетенцій на рівні нижче базового, проявляється недостатність знань, умінь, навичок.
від 0 до 40	«не зараховано» / «незадовільно»	Дисциплінарні компетенції не формувати. Виявляється повне або практично повна відсутність знань, умінь, навичок.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13 Методичне забезпечення

Навчально-методичний комплекс дисципліни у електронному вигляді знаходиться в гугл-класі – Каратанов О. В. Навчально-методичний комплекс дисципліни «Системи контролю версій» [Електронний ресурс] / О. В. Каратанов. – Х. : ХАІ, 2024. – Режим доступу: https://classroom.google.com/c/NjI0ODk3MzgwNTU2?cjc=m5i3jim

14 Рекомендована література

14.1 Базова

1. Git - Основи Git [Електронний ресурс]. – Режим доступу: https://git-scm.com/book/ru/v2
2. GIT HowTo [Електронний ресурс]. — Режим доступу http://githowto.com
3. Git Magic - Preface [Електронний ресурс]. — Режим доступу: http://www-cs-students.stanford.edu/~blynn/gitmagic/intl/ru/
4. Code School - Try Git [Електронний ресурс]. — Режим доступу: http://try.github.io/levels/1/challenges/1
5. git - the simple guide [Електронний ресурс]. — Режим доступу: http://rogerdudler.github.io/git-guide/

14.2 Допоміжна

1. Курс від BitBucket [Електронний ресурс]. — Режим доступу: https://www.atlassian.com/ru/git/tutorials/what-is-version-control

2. Візуалізація команд Git [Електронний ресурс]. — Режим доступу: https://ndpsoftware.com/git-cheatsheet.html