

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
Кафедра математичної інформатики**



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
КОМАНДНА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ /
TEAM SOFTWARE DEVELOPMENT**

для студентів

галузь знань	12 «Інформаційні технології»
спеціальність	122 «Комп'ютерні науки»
освітній рівень	магістр
освітня програма	«Математичні методи штучного інтелекту»
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2022/2023
Семестр	1
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	англійська, українська
Форма заключного контролю	залік

Викладачі: **к.т.н., асистент Олексій ФЕДУРУС** (лекції, лабораторні заняття)

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.

на 20__/20__ н.р. _____ (_____) «__» 20__ р.

КИЇВ – 2022

Розробник: Олексій ФЕДОРУС, к.т.н., асистент кафедри математичної інформатики

ЗАТВЕРДЖЕНО

Зав. кафедри «Математичної інформатики»

Василь ТЕРЕЩЕНКО

Протокол № 1 від «29» серпня 2022 р.

Схвалено гарантом освітньо-наукової програми

«Математичні методи штучного інтелекту»



Ігор ЗАВАДСЬКИЙ

« 31 » серпня 2022 року

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від « 31 » серпня 2022 року № 1.

Голова науково-методичної комісії



Людмила ОМЕЛЬЧУК

« 31 » серпня 2022 року

1. Мета дисципліни «Командна розробка програмного продукту / Team software development» – ознайомлення студентів із сучасними підходами до організації командної розробки додатків та навичок їх практичного застосування.

/

Discipline aim - introducing students to modern approaches to the organization of team development of applications and the skills of their practical application.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. Успішне опанування курсу «Програмування».
2. Володіти сучасними програмними мовами, засобами розробки та проектування програм для розв'язання наукових та прикладних задач та навичками навчальної діяльності.

/

Preliminary demands to master or choice of the course discipline:

1. Course of Programming
2. Know modern programming languages, tools for developing and designing programs for solving scientific and applied problems, and skills in educational activities.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Командна розробка програмного продукту / Team software development» є складовою освітньо-наукової програми підготовки фахівців за другим (магістрським) рівнем вищої освіти *галузі знань* 12 „Інформаційні технології” зі *спеціальності* 122 „Комп'ютерні науки”, *освітньо-професійної програми* „Математичні методи штучного інтелекту”.

Дана дисципліна є обов'язковою навчальною дисципліною за освітньо-науковою програмою “Математичні методи штучного інтелекту”.

Викладається у 1 семестрі 1 курсу в **обсязі 90 год. (3 кредити ECTS)**, в тому числі:

Лекцій – **14 годин.**, лабораторних робіт – **14 год.**, консультації – **2 год.**, самостійна робота - **60 год.**

У курсі передбачено 2 контрольні роботи та 3 лабораторні роботи.

/

Synopsis of the course:

The educational discipline " Team software development" is a component of the educational and scientific program of training specialists at the second (master's) level of higher education in the field of knowledge 12 "Information technologies" from the specialty 122 "Computer science", of the educational and professional program " Informatics".

This subject is a compulsory subject under the educational and scientific program "Informatics".

It is taught in the 1st semester of the 1st course in the amount of 90 hours. (3 ECTS credits), including:

Lectures - 14 hours, laboratory work - 14 hours, consultations - 2 hours, independent work - 60 hours.

The course includes 2 control tasks and 3 laboratory tasks.

4. Завдання (навчальні цілі) / Subject goals:

Основними завданнями дисципліни «Командна розробка програмного продукту / Team software development» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх методологій у командній розробці ПЗ, відповідно до освітньої кваліфікації "Магістр з комп'ютерних наук".

Зокрема, розвивати програмні компетентності (згідно освітньої програми):

- **ЗК02.** Здатність застосовувати знання у практичних ситуаціях.
- **ЗК03.** Здатність спілкуватися державною мовою як усно, так і письмово.
- **ЗК04.** Здатність спілкуватися іноземною мовою.
- **ЗК05.** Здатність вчитися й оволодівати сучасними знаннями.
- **ЗК06.** Здатність бути критичним і самокритичним.
- **ЗК07.** Здатність генерувати нові ідеї (креативність).
- **СК08.** Здатність розробляти і реалізовувати проекти зі створення програмного забезпечення, у тому числі в непередбачуваних умовах, за нечітких вимог та необхідності застосовувати нові стратегічні підходи, використовувати програмні інструменти для організації командної роботи над проектом.
- **СК10.** Здатність оцінювати та забезпечувати якість IT-проектів, інформаційних та комп'ютерних систем різного призначення, застосовувати міжнародні стандарти оцінки якості програмного забезпечення інформаційних та комп'ютерних систем, моделі оцінки зрілості процесів розробки інформаційних та комп'ютерних систем.
- **СК11.** Здатність ініціювати, планувати та реалізовувати процеси розробки інформаційних та комп'ютерних систем та програмного забезпечення, включно з його розробкою, аналізом, тестуванням, системною інтеграцією, впровадженням і супроводом.

/

The main tasks of the discipline "Team software development / Team software development" are the acquisition of knowledge, abilities and skills (competencies) at the level of the latest methodologies in team software development, in accordance with the educational qualification "Master of Computer Science".

Develop program competencies (according to the educational program):

- CC02. Ability to apply knowledge in practical situations.
- CC03. Ability to communicate in the national language both orally and in writing.
- CC04. Ability to communicate in a foreign language.
- CC05. Ability to learn and master modern knowledge.
- CC06. The ability to be critical and self-critical.
- CC07. Ability to generate new ideas (creativity).
- SC08. The ability to develop and implement software creation projects, including in unpredictable conditions, with unclear requirements and the need to apply new strategic approaches, use software tools to organize teamwork on the project.
- SC10. The ability to assess and ensure the quality of IT projects, information and computer systems of various purposes, to apply international standards for assessing the quality of software of information and computer systems, models for assessing the maturity of information and computer systems development processes.
- SC11. Ability to initiate, plan and implement the development processes of information and computer systems and software, including its development, analysis, testing, system integration, implementation and support.

5. Результати навчання за дисципліною / Results of learning::

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати основні поняття та деталі роботи системи контролю версій Git. / Know the basic concepts and details of the Git version control system.	Лекція, лабораторне заняття / Lecture, laboratory classes	Контрольна робота (тест), захист лабораторної роботи / Test, defense of laboratory work	22%
PH1.2	Знати основні методології командної розробки програмних продуктів / Know the basic methodologies of team development of software products	Лекція, лабораторне заняття / Lecture, laboratory classes	Контрольна робота (тест), захист лабораторної роботи / Test, defense of laboratory work	15%
PH1.3	Знати основні підходи до проектування великих систем / Know the main approaches to the design of large systems	Лекція, лабораторне заняття / Lecture, laboratory classes	Контрольна робота (тест), захист лабораторної роботи / Test, defense of laboratory work	15%
PH2.1	Вміти застосовувати на практиці інструментальні програмні засоби системи контролю версій Git. / To be able to apply in practice Git version control system.	Лабораторне заняття, самостійна робота / Lecture, laboratory classes	Захист лабораторної роботи / Defense of laboratory work	30%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програмних систем, скласти письмові звіти / To justify one's own view of the task, to communicate with colleagues on the design and development of software systems, to prepare written reports	Лабораторне заняття / laboratory classes	захист лабораторної роботи / Defense of laboratory work	6%
PH4.1	Організовувати свою самостійну та групову роботу для досягнення результату / Organize independent and group work to achieve results	Самостійна робота, лабораторна робота / Individual work, laboratory classes.	Захист лабораторної роботи / Defense of laboratory work	6%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідаль-	Лабораторна робота / laboratory classes	Захист лабораторної роботи	6%

	ність за їх якість / Be responsible for the performed works, bear responsibility for their quality		/ Defense of laboratory work	
--	--	--	------------------------------	--

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання / Correlation of discipline learning outcomes with program learning outcomes

Результати навчання дисципліни Програмні результати навчання (з опису освітньої програми)	РН 1.1	РН 1.2	РН 1.3	РН 2.1	РН 3.1	РН 4.1	РН 4.2
ПРН 5. Оцінювати результати діяльності команд та колективів у сфері інформаційних технологій, забезпечувати ефективність їх діяльності. / Evaluate the results of teams and collectives in the field of information technologies, ensure the effectiveness of their activities.	+	+	+	+	+	+	+
ПРН 13. Оцінювати та забезпечувати якість інформаційних та комп'ютерних систем різного призначення. / Assess and ensure the quality of information and computer systems for various purposes.	+	+	+	+			
ПРН 14. Тестувати програмне забезпечення. / Test the software.	+	+	+	+			
ПРН 15. Виявляти потреби потенційних замовників щодо автоматизації обробки інформації. / Identify the needs of potential customers regarding the automation of information processing.		+	+				
ПРН 17. Виявляти та усувати проблемні ситуації в процесі експлуатації програмного забезпечення, формулювати завдання для його модифікації або реінжинірингу. / Identify and eliminate problematic situations during software operation, formulate tasks for its modification or reengineering.		+	+				
ПРН 18. Збирати, формалізувати, систематизувати і аналізувати потреби та вимоги до інформаційної або комп'ютерної системи, що розробляється, експлуатується чи супроводжується. / Collect, formalize, systematize and analyze the needs and requirements for the information or computer system being developed, operated or supported.		+	+				

7. Схема формування оцінки / Grading.

7.1. Форми оцінювання студентів / Student evaluation forms:

– семестрове оцінювання:

1. Контрольна робота 1 / Test 1: PH1.1 – 20 балів/12 балів
2. Контрольна робота 2 / Test 2 : PH1.2, PH1.3 – 20 балів/12 балів;
3. Лабораторна робота 1 (проект) / Laboratory work 1: PH1.1, PH2.1, PH3.1, PH4.1, PH4.2 – 20 балів/12 балів
4. Лабораторна робота 2 (проект) / Laboratory work 2: PH1.2, PH2.1, PH3.1, PH4.1, PH4.2 – 20 балів/12 балів.
5. Лабораторна робота 3 (проект) / Laboratory work 3: PH1.3, PH2.1, PH3.1, PH4.1, PH4.2 – 20 балів/12 балів.

Підсумкове оцінювання (у формі заліку):

- залікові бали визначаються як сума оцінок/балів за всіма успішно оціненими результатами навчання, передбаченими даною програмою;
- оцінки нижче від мінімального порогового рівня не додаються;
- мінімальний пороговий рівень для сумарної оцінки за всіма компонентами становить 60% від максимально можливої кількості балів.

Final assessment (in the form of credit):

- credits are defined as the sum of grades/points for all successfully assessed learning outcomes provided in this program.
- grades below the minimum threshold level are not added;
- the minimum threshold level for the total assessment for all components is 60% of the maximum possible number of points.

7.2 Організація оцінювання/ Evaluation organization:

Терміни проведення форм оцінювання/ Deadlines for evaluation forms:

1. Контрольна робота 1 / test 1: до 7 тижня семестру / before week 7
2. Контрольна робота 2 / test 2: до 14 тижня семестру / before week 14
3. Лабораторна робота 1 / Laboratory work 1: до 5 тижня семестру. / before week 5
4. Лабораторна робота 2 / Laboratory work 2: до 10 тижня семестру. / before week 10
5. Лабораторна робота 3 / Laboratory work 3: до 14 тижня семестру. / before week 14

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перескладання контрольних робіт здійснюються у відповідності до «Положення про організацію освітнього процесу» від 07.05.2018 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

The student has the right to one retake of each test with the possibility of receiving a maximum of 80% of the points originally determined for this test. The deadline for rewriting is determined by the teacher.

In the case of a student's absence due to valid reasons, practice and retaking of test papers are carried out in accordance with the "Regulations on the Organization of the Educational Process".

In case of low-quality performance of laboratory work, the teacher has the right not to credit the laboratory work, or to reduce the points for it.

A student has the right to submit laboratory work after the deadline set for them, but with the loss of one point for each week that has passed since the deadline for its submission.

7.3 Шкала відповідності оцінок / Rating correspondence scale

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни

Тематичний план лекцій та лабораторних занять

№ лекції	Назва лекції / Lecture topic	Кількість годин / Hours		
		Лекції/ Lectures	Лаборат. Занять / lab. work	Сам. р-та / indiv. work
Частина 1. Системи контролю версій Part 1. Version control systems				
1.	Тема 1. Вступ. Командна розробка. Основні задачі. Самостійна робота. Опрацювання лекційного матеріалу. Topic 1. Introduction. Team development. Main tasks. Individual work. Processing of lecture material.	2		6
2.	Тема 2. Розвиток систем контролю версій Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 2. Development of version control systems Individual work. Processing of lecture material. Performing laboratory work.	2	1	8
3.	Тема 3. Особливості роботи системи контролю версій Git. Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 3. Peculiarities of the Git version control system. Individual work. Processing of lecture material. Performing laboratory work.	2	1	6
	<i>Контрольна робота 1 / Test 1</i>		1	
Контроль за підсумками лабораторної роботи 1 / Control part 1				
Всього по частині 1 / Part 1 in total		6	3	20
Частина 2. Процес розробки програмного забезпечення Part 2. Software development process				
5.	Тема 4. Розробка ПЗ за допомогою каскадної моделі. V-Model. Dual Vee Model. Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 4. Software development using the cascade model. V-Model. Dual Vee Model. Individual work. Processing of lecture material. Performing laboratory work.	1	1	6
6.	Тема 5. Розробка ПЗ за допомогою ітеративної моделі Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 5. Software development using an iterative model Individual work. Processing of lecture material. Performing laboratory work.	1	1	4
7.	Тема 6. Розробка ПЗ за допомогою спіральної моделі Самостійна робота. Опрацювання лекційного матеріалу.	1	1	4

	Виконання лабораторної роботи. Topic 6. Software development using a spiral model Individual work. Processing of lecture material. Performing laboratory work.			
8.	Тема 7. Гнучка(Agile) модель розробки ПЗ Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 7. Agile software development model Individual work. Processing of lecture material. Performing laboratory work.	4	3	10
9.	Тема 8. Техніки що застосовуються при використанні Agile-моделі Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 8. Techniques used when using the Agile model Individual work. Processing of lecture material. Performing laboratory work.	2	1	6
10.	Тема 9. Техніки для проектування ПЗ. Самостійна робота. Опрацювання лекційного матеріалу. Виконання лабораторної роботи. Topic 9. Software design techniques. Individual work. Processing of lecture material. Performing laboratory work.	1	1	4
	<i>Контрольна робота 2 / test 2</i>		1	
	Контроль за підсумками лабораторних робіт 2 та 3 / laboratory work 2 and 3 submission			
	Всього по частині 2 / Part 2 total	10	9	40
	ВСЬОГО / TOTAL	16	12	60
	Консультації / Consultations	2		

Загальний обсяг 90 годин, в тому числі:

Лекцій – **16 годин**.

Лабораторних – **12 год**.

Консультації – **2 год**.

Самостійна робота - **60 год**.

In total: 90 hours:

Lectures – **16 hours**

Laboratory lessons – **12 hours**

Consultations – **2 hours**

Individual work - **60 hours**.

9. Рекомендовані джерела / References:

Основні / Main

1. Software Engineering Body of Knowledge (SWEBOK) - Link: <https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3>
2. Braude J. Eric. Software Engineering Modern Approaches Second Edition - Waveland Press, Inc., 2011- Link: <http://ndl.ethernet.edu.et/handle/123456789/26455>
3. Ivar Jacobson et al. The Essentials of Modern Software Engineering. – ACM, 2019 – Link: https://www.academia.edu/43422359/The_Essentials_of_Modern_Software_Engineering
4. Scott Chacon, Ben Straub. Pro Git – Apress, 2021 – Link: <https://git-scm.com/book/en/v2>