

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«__» _____ 2019 року

**РОБОЧА ПРОГРАМА ВИРОБНИЧОЇ ПРАКТИКИ
РОЗРОБКА ПРОГРАМНО-
ІНФОРМАЦІЙНИХ СИСТЕМ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Програмне забезпечення систем
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2019/2020
Семестр	4
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	диференційований залік

Керівник практики від кафедри: **к. т. н., доцент Демківський Є.О.**

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

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

Розробник: Демківський Євген Олександрович, к. т. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

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

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Мета практики – ознайомлення студентів з актуальними практичними підходами до тестування і забезпечення якості програмного забезпечення.

2. Попередні вимоги: відсутні.

3. Анотація виробничої практики. Виробнича практика «Розробка програмно-інформаційних систем» є складовою програми підготовки фахівців за другим (магістерським) рівнем вищої освіти галузі знань 12 Інформаційні технології зі спеціальності 121 Інженерія програмного забезпечення, освітньо-наукової програми «Програмне забезпечення систем».

Виробнича практика проводиться у **4** в обсязі – **120 год. (4 кредити ECTS)**.

Організація виробничої практики. Навчально-методичне керівництво з урахуванням видів практик здійснює кафедра інтелектуальних програмних систем. Відповідальний за проведення практики вчасно доводить до студента інформацію щодо баз практики. Студенти у зазначений термін подають на кафедру на ім'я завідуючого заяву з зазначенням бази практики, що обрана. Офіційною підставою для проведення виробничої практики студентів на виробництві є договір, який укладається між ЗВО та підприємством. Договори укладають з підприємствами за заявками фахової кафедри інтелектуальних програмних систем. Порядок надання заявок та укладання договорів установлюється наказом ректора та розпорядженням декана. Керівник підприємства-базы практики видає наказ щодо практики, де визначається порядок організації та проведення практики, заходи для створення необхідних умов студентам-практикантам задля виконання ними програми практики, по охороні праці та запобіганню виникнення нещасних випадків: по контролю за виконанням студентами правил внутрішнього трудового розпорядку, інші заходи для проведення практики у відповідності до Положення про проходження виробничої практики студентами ЗВО, призначає керівника практики від підприємства. Перед початком практики кафедра проводить виробничу нараду студентів-практикантів та викладачів – керівників практики для роз'яснення мети, змісту та порядку проходження практики. Перед відправкою до бази практики студент повинен одержати направлення, щоденник практики, програму її проходження, індивідуальне завдання. Коли студент прибуває на практику, він за перші три доби повинен оформити сповіщення про прибуття до підприємства та про початок виробничої практики. Доки не отримано таке сповіщення студент вважається як той, який не з'явився на практику, що розглядається як факт порушення навчального процесу. Студенти, які не пройшли практику за поважними причинами, направляються на практику у строки визначені деканатом. У період проходження практики студент повинен:

- виконувати завдання, передбачені програмою практики та календарним графіком;
- підпорядковуватися діючим правилам внутрішнього трудового розпорядку підприємства;
- суворо дотримуватися правил техніки безпеки та охорони праці;
- працювати на робочому місці, яке вказано керівником практики від підприємства і нести відповідальність за виконану роботу та її результати нарівні зі штатними робітниками;
- систематично вести щоденник проходження практики.

Бази практики. Виробнича практика може проводитися як на базі кафедри інтелектуальних програмних систем, так і на базі підприємств, організацій, науково-дослідницьких інститутів, банків, страхових компаній та інших установ, що займаються проектуванням, розробкою, впровадженням та експлуатацією автоматизованих інформаційних систем. Вибір баз практики здійснюється за погодженням з кафедрою

інтелектуальних програмних систем з урахуванням завдань практики та можливості їх реалізації.

4. Завдання (навчальні цілі). Виробнича практика покликана сформувати у студента професійні вміння, навички прийняття самостійних рішень на конкретній ділянці роботи в реальних, виробничих умовах шляхом виконання обов'язків, властивих його майбутній професійній та організаційно-управлінській діяльності. Поставлені цілі реалізують шляхом самостійного вивчення виробництва й виконання кожним студентом, в умовах підприємства, передбачених програмою окремих виробничих й організаційно-управлінських завдань. Зокрема, розвивати:

- Здатність спілкуватися іноземною мовою як усно, так і письмово (ЗК-2).
- Здатність мотивувати людей та рухатися до спільної мети, працювати в команді співробітників (ЗК-4).
- Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності) (ЗК-5).
- Здатність удосконалювати свої навички на основі аналізу попереднього досвіду (ЗК-6).
- Здатність приймати обґрунтовані рішення (ЗК-8).
- Здатність аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення (СК-1).
- Здатність ідентифікувати, класифікувати та описувати проектні завдання, знаходити раціональні методи й підходи до їх розв'язання (СК-2).
- Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК-3).
- Здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (СК-4).
- Здатність оцінювати ступінь обґрунтованості застосування специфікацій, стандартів, правил і рекомендацій в професійній галузі та дотримуватися їх при реалізації процесів життєвого циклу програмного забезпечення (СК-5).
- Здатність ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами (СК-6).
- Здатність систематизувати професійні знання щодо створення і супроводження програмного забезпечення (СК-7).
- Здатність розробляти і координувати процеси, фази та ітерації життєвого циклу програмних систем на основі застосування відповідних моделей, методів та технологій розробки програмного забезпечення (СК-8).

5. Результати навчання за дисципліною.

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			
РН1.1	Знати основні етапи процесу проектування ПЗ.	Самостійна робота.	Поточне оцінювання.	12%
РН1.2	Знати призначення, можливості і технології розробки та опису розробленого ПЗ.	Самостійна робота.	Поточне оцінювання.	15%
РН1.3	Знати принципи процедурного і структурного програмування, базові типи даних мови, оператори управління програмою.	Самостійна робота.	Поточне оцінювання.	15%
РН2.1	Вміти розробляти додатки з використанням принципів процедурного і структурного програмування.	Самостійна робота.	Поточне оцінювання.	20%
РН2.2	Вміти самостійно тестувати ПЗ як вручну, так і за допомогою сучасних інструментальних засобів автоматизації тестування.	Самостійна робота.	Поточне оцінювання.	20%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань розробки та тестування ПЗ, складати письмові звіти.	Самостійна робота.	Поточне оцінювання.	5%
РН4.1	Організувати свою самостійну роботу для досягнення результату.	Самостійна робота.	Поточне оцінювання.	8%
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість.	Самостійна робота.	Поточне оцінювання.	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН1.1	РН1.2	РН1.3	РН2.1	РН2.2	РН3.1	РН4.1	РН4.2
	Програмні результати навчання							
ПРН-1. Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення.	+							
ПРН-2. Обґрунтовувати вибір методів формування вимог до програмної системи, розробляти, аналізувати та систематизувати вимоги			+			+		
ПРН-3. Знати і застосовувати базові концепції і методології моделювання інформаційних процесів.	+			+				
ПРН-4. Оцінювати і обирати методи і моделі розробки, впровадження, експлуатації програмних засобів та управління ними на всіх етапах життєвого циклу.			+	+				
ПРН-5. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.			+					
ПРН-6. Аналізувати, оцінювати і обирати методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії.				+	+			
ПРН-7. Обґрунтовано обирати парадигми і мови програмування для вирішення прикладних завдань; застосовувати на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення.				+	+			
ПРН-8. Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення.					+			
ПРН-9. Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення.		+			+	+		
ПРН-10. Вміти приймати організаційно-управлінські рішення в умовах невизначеності.							+	
ПРН-11. Набувати нові наукові і професійні знання, вдосконалювати навички, прогнозувати розвиток програмних систем та інформаційних технологій.							+	+

7. Схема формування оцінки.

Керівництво проходженням практики здійснює кафедра інтелектуальних програмних систем. Викладач кафедри, відповідальний за проведення практики:

- забезпечує якісне виконання програми практики та високу якість її проведення;
- призначає керівниками виробничої практики досвідчених викладачів;
- розподіляє на основі укладених з підприємством договорів студентів за базами практики;
- призначає старшого з групи студентів, які проходять практику на одному підприємстві;
- забезпечує підприємство, а також самих практикантів програмами практики;
- здійснює суворий контроль за організацією та проведенням виробничої практики студентів на підприємстві, і дотриманням строків та змісту.

Обов'язки призначеного кафедрою керівника практики від університету:

- забезпечити проведення всіх організаційних заходів перед відправкою студентів на практику;
- забезпечити високу якість проходження практики і сувору відповідальність її навчальному плану й програмі;
- надавати консультації студентам з усіх питань практики;
- контролювати додержання студентами-практикантами правил внутрішнього розпорядку;
- керувати науково-дослідною роботою студентів, яка передбачена завданнями кафедри;
- здійснювати поточний контроль проходження практики у відповідності із календарним графіком;
- розглядати звіти студентів з практики, надавати відгук та висновок з практики та звіту;
- подавати письмовий звіт про проходження практики, надавати пропозиції та зауваження з досконалості практичної підготовки студентів.

В обов'язки керівника практики від підприємства входить:

- організувати проходження практики закріплених за ним студентів у тісному контакті з керівником від університету;
- познайомити студентів з організацією праці на конкретному робочому місці;
- здійснювати постійний контроль за виробничою роботою практикантів, допомагати їм вірно виконувати всі завдання на даному робочому місці, консультувати щодо виробничих питань;
- контролювати ведення щоденників, підготовку звітів студентами-практикантами та складати на кожного студента виробничу характеристику-відгук керівника практики від підприємства, який заноситься до відповідного розділу щоденника виробничої практики;
- ознайомитися зі звітом студента та дати оцінку звіту і роботі студента;
- прийняти участь у захисті звітів студентами в якості експерта та екзаменатора.

7.1. Форми оцінювання студентів.

Оформлення та захист звіту. У ході практики студент повинен скласти письмовий звіт, підписати його у керівника практики від підприємства, поставити печатку і разом із оформленим відповідним чином щоденником практики, характеристикою-відгуком від підприємства здати керівнику практики від університету. Звіт з практики складається після виконання розділів програми та опрацювання матеріалів. Його оформлення закінчується на підприємстві до моменту закінчення практики. Він містить інформацію про вид практики, тему індивідуального завдання, виконавця звіту та керівника практики. Завдання на практику містить інформацію про назву задачі (модуля), строк виконання індивідуального завдання, вхідні дані для вирішення задач індивідуального завдання, перелік графічного матеріалу, підпис керівника від ЗВО та студента. На сторінці підсумків ставиться дата оформлення та підпис студента. Захист звіту з оцінкою здійснюється комісією, яка організовується на кафедрі інтелектуальних програмних систем (PH1.1, PH1.2, PH1.3, PH2.1, PH2.2, PH3.1, PH4.1, PH 4.2).

7.2 Організація оцінювання.

Терміни проведення форм оцінювання:

1. Індивідуальний звіт про проходження практики: по завершенню практики.

Оцінка визначається як сума усіх балів по усіх компонентах передбачених програмою дисципліни.

Студенти можуть бути недопущені до заліку, якщо під час практики вони набрали менше ніж 60 балів.

7.3. Шкала відповідності оцінок.

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

8. Структура виробничої практики. Тематичний план.

№ лекції	Зміст робіт	Кількість годин
1	Проходження інструктажу з техніки безпеки.	2
2	Ознайомлення з метою та програмою практики, отримання завдання.	4
3	Специфікація програмних вимог.	16
4	Написання програмного коду.	36
5	Тестування програмного коду.	16
6	Отримання та аналіз результатів.	14

7	Оформлення документів згідно зі стандартами ДСТУ.	8
8	Створення презентацій засобами PowerPoint.	8
9	Оформлення звіту згідно з ДСТУ.	16
ВСЬОГО		120

Загальний обсяг – **120** год.

9. Рекомендовані джерела:

Основні:

1. Beizer, Boris. Software Testing Techniques, 2nd edition, Boston, MA.: International Thomson Computer Press, 1990 (ISBN 1850328803).
2. Chrissis, M. B.; Konrad, M.; & Shrum, S. CMMI: Guidelines for Process Integration and Product Improvement. Boston, MA: Addison-Wesley, 2003 (ISBN 0321154967).
3. Cote, Marc-Alexis; Suryn, Witold; Martin, Robert A.; & Laporte, Claude Y. "Evolving a Corporate Software Quality Assessment Exercise: A Migration Path to ISO/IEC 9126." Software Quality Engineering 6, 3 (2004).
4. Fenton, Norman E. & Pfleeger, Sharon L. Software Metrics: A Rigorous and Practical Approach, 2nd ed. Boston, MA: International Thomson Computer Press, 1996 (ISBN 1850322759).
5. Grady, Robert B. Practical Software Metrics for Project Management and Process Improvement. Englewood Cliffs, NJ: Prentice Hall, 1992 (ISBN 0137203845).
6. Halstead, Maurice. H. Elements of Software Science. New York, NY: Elsevier, 1977 (ISBN 0444002057).
7. Humphrey, Watts S. Managing the Software Process. Reading, MA: Addison Wesley, 1989 (ISBN 0201180952).
8. Humphrey, Watts S. A Discipline for Software Engineering. Reading, MA: Addison-Wesley, 1995 (ISBN 0201546108).
9. Humphrey, Watts S. Introduction to the Team Software Process. Reading, MA: Addison-Wesley, 2000 (ISBN 020147719X).
10. ISO. ISO/IEC 15939:2002, Software engineering – Software Measurement Process. Geneva, Switzerland: International Organization for Standardization, 2002.
11. ISO. ISO/IEC 9126-1:2001: Software Engineering – Product Quality. Part 1: Quality Model. Geneva, Switzerland: International Organization for Standardization, 2001.
12. ISO. ISO/IEC TR 9126-2:2003: Software Engineering – Product Quality. Part 2: External Metrics. Geneva, Switzerland: International Organization for Standardization, 2003.
13. ISO. ISO/IEC TR 9126-3:2003: Software Engineering – Product Quality. Part 3: Internal Metrics. Geneva, Switzerland: International Organization for Standardization, 2003.
14. ISO. ISO/IEC TR 9126-4:2004: Software Engineering – Product Quality. Part 4: Quality in Use Metrics. Geneva, Switzerland: International Organization for Standardization, 2004.
15. Kan, Stephen H. Metrics and Models in Software Quality Engineering, 2nd ed. Boston, MA: Addison-Wesley, 2003 (ISBN 0201729156).

16. McGarry, John; Card, David; Jones, Cheryl; Layman, Beth; Clark, Elizabeth; Dean, Joseph; & Hall, Fred. Practice Software Measurement: Objective Information for Decision Makers, Boston, MA: Addison-Wesley, 2002 (ISBN 0201715163).

Додаткові:

1. Basili, Victor R. "Quantitative Software Complexity Models: A Panel Summary." IEEE Proceedings of the Workshop on Quantitative Software Models for Reliability, Complexity, and Cost. October 1979.
2. Basili, Victor R. & Weiss, David M. "A Methodology for Collecting Valid Software Engineering Data." IEEE Transactions on Software Engineering 10, 6 (November 1984): 728-738.
3. Fagan, Michael E. "Design and code inspections to reduce errors in program development." IBM Systems Journal 38, 2 & 3 (1999): 258-287.
4. Fenton, Norman. E. & Ohlsson, Niclas. "Quantitative Analysis of Faults and Failures in a Complex Software System." IEEE Transactions on Software Engineering 26, 8 (August 2000): 797-814.
5. Florac, W. Software Quality Measurement: A Framework for Counting Problems and Defects (CMU/SEI-92-TR-022, ADA258556). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1992.
6. McCabe, T. "A Complexity Measure." IEEE Transactions on Software Engineering 2, 4 (December 1976): 308-320.
7. McGraw, Gary. "Software Security." IEEE Security and Privacy 2, 2 (March-April 2004): 80-83.