НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ ТА ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Кафедра комп'ютерних наук

«ЗАТВЕРДЖУЮ»
декан факультету інформаційних технологій
_____ Глазунова О.Г.
«25» ____05_____ 2022 р.

«СХВАЛЕНО»
на засіданні кафедри комп'ютерних наук
Протокол № ____ від «___» _____ 20____ р.
Завідувач кафедри
_____ Б. Л. Голуб

«РОЗГЛЯНУТО»
Гарант ОП 121 «Інженерія програмного забезпечення»
гарант ОП
_____ Лялецький О.В.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА
«Інженерія програмного забезпечення»

за спеціальністю 121 «Інженерія програмного забезпечення»
галузі знань 12 «Інформаційні технології»

Факультет інформаційних технологій

Розробник: ст. викладач кафедри комп'ютерних наук Панкратьєв В.О.

Київ – 2022 р.

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмної інженерії

(назва)

Галузь знань, напрям підготовки, спеціальність, освітньо-кваліфікаційний рівень		
Галузь знань	12 «Інформаційні технології»	
Спеціальність	121 – «Інженерія програмного забезпечення»	
Освітня програма	«Інженерія програмного забезпечення»	
Освітній ступінь	Бакалавр	
Характеристика навчальної дисципліни		
Вид	Обов'язкова	
Загальна кількість годин	120	
Кількість кредитів ECTS	3	
Кількість змістових модулів	2	
Курсовий проект (робота) (за наявності)		
Форма контролю	*Іспит*	
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Рік підготовки (курс)	1	
Семестр	1	
Лекційні заняття	*15 год.*	
Практичні, семінарські заняття	-	
Лабораторні заняття	*30 год.*	
Самостійна робота	*75 год.*	
Індивідуальні завдання	-	
Кількість тижневих аудиторних годин для денної форми навчання	*4 год.*	

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою дисципліни «Основи програмної інженерії» є вивчення сучасних інженерних принципів (методів) створення надійного, якісного програмного забезпечення, яке задовольняє пропонованим до нього вимогам; формування у студентів розуміння необхідності застосування даних принципів програмної інженерії та набуття студентами теоретичних знань та практичних навичок з проектування програмних продуктів.

Завданням дисципліни вивчення дисципліни «Основи програмної інженерії» є:

– формування у студентів загального розуміння програмної інженерії як галузі наукового і практичного знання, актуальності програмної інженерії як дисципліни, що охоплює всі аспекти розробки програмного забезпечення;

– поглиблення і розширення поняття життєвий цикл програмного забезпечення (ЖЦ ПЗ), детальний розгляд всіх етапів ЖЦ ПЗ;

–розуміння необхідності всіх стадій в ЖЦ ПЗ;

– отримання початкових умінь і навичок в галузі розробки ПЗ з проходженням всіх стадій ЖЦ ПЗ зі складанням програмної документації;

–розвиток логічного і алгоритмічного мислення студентів;

– вивчення студентами техніко-економічних і організаційних питань проектування програмних засобів;

У результаті вивчення навчальної дисципліни студент повинен

знати:

– переваги інженерного підходу до створення програмного забезпечення;

– основні складнощі, що виникають при впровадженні такого підходу;

– зв'язок програмної інженерії з життєвим циклом програмних засобів;

– основні етапи і зміст робіт на кожному етапі розробки програмних систем;

– основні поняття і моделі процесу розробки ПЗ - проведення планування, аналіз системи, аналіз предметної області та вимог до створюваної системи.

вміти:

- орієнтуватися в різних способах організації і моделях процесу розробки: каскадна модель, ітеративна модель, модель покрокової розробки, спіральна модель та ін.

- застосовувати отримані знання у своїй професійній діяльності.

Загальні компетентності (ЗК)

К05. Здатність вчитися і оволодівати сучасними знаннями.

Фахові компетентності спеціальності (ФК)

К14. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

К23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

К25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення, враховуючи специфіку природоохоронної галузі та сільського господарства

3. ПРОГРАМА ТА СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

–повного терміну денної форми навчання;

Змістовний модуль №1. Предмет програмної інженерії. Життєвий цикл розробки програмного забезпечення.

Тема 1. Предмет програмної інженерії

Поняття про програмну інженерію. Програмна інженерія та програмування.

Тема 2. Життєвий цикл розробки програмного забезпечення

Процес розробки програмного забезпечення. Парадигма розробки програмного забезпечення. Моделі.

Тема 3. Гнучкі методи розробки

Ітеративні та інкрементні моделі. Життєвий цикл програмного забезпечення по RAD. Методологія RUP.

Тема 4. Архітектура ПЗ

Поняття про архітектуру ПЗ. Складові елементи архітектури ПЗ. Архітектура інформаційних систем.

Тема 5. Вимоги до програмного забезпечення

Види вимог. Функціональні і нефункціональні вимоги. Системні вимоги. Розробка вимог. Управління вимогами.

Змістовний модуль №2. Структурні елементи інтерфейсу користувача. Проектування та реалізація користувацьких інтерфейсів.

Тема 1. Технічне завдання

Структурні елементи інтерфейсу користувача. Атрибути та методи (івенти) елементів інтерфейсу. Поняття меню, його структура.

Тема 2. Управління ризиками

Визначення ризиків. Аналіз ризиків. Моніторинг ризиків. Планування ризиків.

Тема 3. Тестування, сертифікація, верифікація ПЗ

Стандартизація якості. Методи забезпечення якості ПЗ. Поняття про тестування. Види тестування. Тестування чорної скриньки. Тестування білої скриньки. Робота з помилками. Сертифікація. Верифікація.

Назви змістових модулів і тем	Кількість годин												
	денна форма							Заочна форма					
	тижні	усього	у тому числі					усього	у тому числі				
			л	п	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13	14
Змістовий модуль 1. Життєвий цикл програмного продукту													
Тема 1. Предмет програмної інженерії	1		2		4								
Тема 2. Життєвий цикл розробки програмного забезпечення	2-3		4		8								
Тема 3. Гнучкі методи розробки	4		2		4								
Тема 4. Архітектура ПЗ	5		2		4								
Разом за змістовим модулем 1	5		10		20								
Змістовий модуль 2. Вимоги до програмного забезпечення													
Тема 1. Вимоги до програмного забезпечення	6		4		8								
Тема 2. Технічне завдання	7		2		4								
Тема 3. Управління ризиками	8		1		2								
Разом за змістовим модулем 2	3		5		10								

Усього годин	16	15	5	30								
Курсовий проект (робота) з _____ _____ (якщо є в робочому навчальному плані)		-	-	-		-		-	-	-		-
Усього годин	45	15		30								

4. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин
1	Збір і аналіз інформації про предметну область	4
2	Етапи розробки програмного проекту. Парадигма розробки ПЗ. Вибір моделі життєвого циклу.	12
3	Вибір архітектури ІС	4
4	Визначення вимог	4
5	Розробка технічного завдання	4
6	Аналіз ризиків	2
		30

5. КОНТРОЛЬНІ ПИТАННЯ, КОМПЛЕКТИ ТЕСТІВ ДЛЯ ВИЗНАЧЕННЯ РІВНЯ ЗАСВОЄННЯ ЗНАНЬ СТУДЕНТАМИ.

1. Легкість застосування програмного забезпечення це:
а) характеристики ПЗ, що дозволяють мінімізувати зусилля користувача з підготовки вихідних даних, застосування ПО;
б) відношення рівня послуг, що надаються ПО користувачеві при заданих умовах, до обсягу використовуваних ресурсів;
в) характеристики ПЗ, що дозволяють мінімізувати зусилля по внесенню змін для усунення в ньому помилок і по його модифікації.

2. Мобільність програмного забезпечення це:
а) здатність ПО виконувати набір функцій, які задовольняють потреби користувачів;
б) здатність ПС безвідмовно виконувати певні функції при заданих умовах протягом заданого періоду часу;
в) здатність ПО бути перенесеним з одного середовища (апаратного / програмного) в інше.

3. Вкажіть правильну послідовність етапів при каскадної моделі життєвого циклу: а) Визначення вимог -> Тестування -> Реалізація; б) Проектування -> Реалізація -> Тестування; в) Проектування -> Визначення вимог -> Реалізація.

4 Стійкість програмного забезпечення - це:
а) властивість, що характеризує здатність ПС завершувати автоматично коректне функціонування ПК, незважаючи на неправильні (помилкові) вхідні дані;
б) властивість, здатна протистояти навмисним або ненавмисним деструктивним діям користувача;
в) властивість, що характеризує здатність ПС продовжувати коректне функціонування, незважаючи на неправильні (помилкові) вхідні дані.

5. UML - це:
а) мова програмування, що має синтаксис схожий з C ++;
б) уніфікована мова візуального моделювання, використовує нотацію діаграм; в) набір стандартів і специфікацій якості програмного забезпечення.

6. UML - це:
а) універсальна багатоваріантна мова;
б) універсальна багатонаціональна мова;
в) уніфікована мова моделювання.

7. При конструюванні програмного забезпечення на етапі розробки або вибору алгоритму рішення реалізується наступне:
а) архітектурне обробки програми;
б) вибір мови програмування; в) вдосконалення програми.

8. Проектування ПО в основному розглядається як
а) архітектурне проектування;
б) комунікаційні методи;
в) детальні методи.

9. На етапі тестування користувач виконує наступне: а) синтаксичне налагодження; б) вибір тестів і методу тестування;
в) визначення форми видачі результатів.

10. Що з наведеного не є одним з методів проектування програмного забезпечення? а) структурне програмування; б) об'єктно-орієнтоване програмування; в) алгебраїчне програмування.

11. Як називається процес розбиття однієї складної задачі на кілька простих підзадач? а) абстракція; б) декомпозиція; в) реінжиніринг.

12. Що з наведеного є критеріями оцінки зручності інтерфейсів?
а) швидкість навчання;
б) адаптація до стилю роботи користувача;
в) всі відповіді правильні.

13. Інтерфейс користувача - це
а) набір методів взаємодії комп'ютерної програми і користувача цієї програми; б) набір методів для взаємодії між програмами; в) спосіб взаємодії між об'єктами.

14. Інтерфейс-це
а) перш за все, набір правил;
б) набір завдань користувача, які він вирішує за допомогою системи; в) спосіб взаємодії між об'єктами.

15. Технічне завдання - це
а) документ пояснень для замовника;
б) вихідний документ для здачі ПЗ в експлуатацію;
в) вихідний документ для проектування, розробки автоматизованої системи.

16. Аналіз вимог -
а) відображення функцій системи і її обмежень в моделі проблеми;
б) показник супроводжуваності, який визначає необхідні зусилля для діагностики випадків відмов;
в) відображення частин програм, які будуть модифікуватися.

17. Архітектура програмної системи -
а) декомпозиція рішення для виділеного спектра завдань домену на підсистеми або ієрархію підсистем;
б) визначення системи в термінах обчислювальних складових (підсистем) і інтерфейсів між ними, яке відображає правила декомпозиції проблеми на складові; в) відповідні варіації складу виділених компонент.

18. Головними областями програмної інженерії не є: а) процес інженерії ПС; б) управління конфігурацією; в) конструювання ПО; г) інженерія вимог.

19. Якість ПЗ - це:
а) набір властивостей продукту, які характеризують його здатність задовольнити встановлені або передбачувані потреби замовника;
б) ступінь автоматизованого виконання завдань процесів життєвого циклу;
в) вартість робіт з проектування та розробки ПЗ.

20. Валідація -
а) забезпечення відповідності розробки вимогам її замовників;
б) перевірка правильності трансформації проекту в код реалізації;
в) виявлення всіх помилок.

21. Верифікація -
а) забезпечення відповідності розробки вимогам її замовників; б) перевірка правильності трансформації проекту в програму;
в) дії на кожній стадії життєвого циклу з перевірки і підтвердження відповідності стандартам.

22. В обговоренні вимог на систему беруть участь:
а) аналітики і розробники майбутньої системи;
б) представники замовника з декількох професійних груп; в) фахівці, що проводять інсталяцію системи.

23. Проектування ПО - це:
а) заходи щодо аналізу сформульованих у вимогах атрибутів якості, оцінки різних аспектів ПО;

б) процес визначення архітектури, компонентів, інтерфейсів, інших характеристик системи і кінцевого складу програмного продукту;
в) створення працюючого ПО з залученням методів верифікації, кодування і тестування компонентів.

24. Розробка вимог включає в себе наступні основні розділи:
а) аналіз вимог; б) систематизація вимог; в) збір вимог;

г) управління вимогами.

25. До процесу розробки ПО включає такі процеси:
а) супроводу;
б) проектування;
в) експлуатація.

26. Послідовність робіт по каскадної моделлю:
а) вимоги, проектування, реалізація; б)
проектування, супровід, тестування; в) вимоги,
супровід, тестування.

27. Проектування -
а) перетворення вимог в послідовність проектних рішень по системі; б)
визначення головних структурних особливостей системи;
в) визначення подробиць функціонування і зв'язків для всіх компонент системи.

28. Модель життєвого циклу -
а) визначення певних дій, які супроводжують зміни станів об'єктів;
б) типова схема послідовності робіт на етапах розробки програмного продукту;
в) відображення динаміки змін стану кожного класу об'єктів.

29. Зрозумілість - це
а) атрибут функціональності, який вказує на можливість запобігати несанкціонований доступ;
б) атрибут надійності, який вказує на здатність програми до перезапуску для повторного виконання;
в) атрибут зручності, що визначає зусилля, необхідні для розпізнавання логічних концепцій та умов їх застосування.

30. Архітектура системи - це:
а) структурна схема компонентів системи, що взаємодіють між собою через інтерфейси;
б) структурна схема компонентів системи, які не взаємодіють між собою;
в) структурна схема інтерфейсів системи, що взаємодіють між собою через компоненти.

6. МЕТОДИ НАВЧАННЯ

Передбачено проведення занять у вигляді лекцій, лабораторних та самостійних робіт. На лекціях застосовуються мультимедійні засоби та дискусії. Лабораторні роботи проводяться у двох частинах – пояснення завдання та захист студентами їхнього виконання. Самостійна робота може бути проведена за вказівкою викладача або за вибором студента відкритого курсу та демонстрації сертифікату проходження цього курсу. Для розповсюдження усіх матеріалів як збоку викладача, так і з боку студента, використовується платформа *moodle*, що розташована за посиланням *elearn.nubip.ua*.

7. ФОРМИ КОНТРОЛЮ

Для забезпечення контролю викладач оцінює виконання лабораторних та самостійних робіт і тестування. Лабораторні роботи мають бути захищені з демонстрацією виконаного завдання та відповідями на запитання викладача. Самостійні роботи захищаються доповідями з презентацією. Курс розбитий на 2 модулі. По завершенню кожного модуля необхідно скласти тест, а по завершенню усього курсу – екзамен.

Розподіл балів, які отримують студенти. Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамени та заліки у НУБіП України» (наказ про уведення в дію від 27.12.2019 р. No 1371)

Оцінка національна	Оцінка ЄКТС	Визначення оцінки ЄКТС	Рейтинг студента, бали
Відмінно	**A**	**ВІДМІННО** – відмінне виконання лише з незначною кількістю помилок	**90 – 100**
Добре	**B**	**ДУЖЕ ДОБРЕ** – вище середнього рівня з кількома помилками	**82–89**
	C	**ДОБРЕ** – в загальному правильна робота з певною кількістю грубих помилок	**74–81**
Задовільно	**D**	**ЗАДОВІЛЬНО** – непогано, але зі значною кількістю недоліків	**64–73**
	E	**ДОСТАТНЬО** – виконання задовольняє мінімальні критерії	**60–63**
Незадовільно	**FX**	**НЕЗАДОВІЛЬНО** – потрібно працювати перед тим, як отримати залік (позитивну оцінку)	**35–59**
	F	**НЕЗАДОВІЛЬНО** – необхідна серйозна подальша робота	**01–34**

Для визначення рейтингу студента (слухача) із засвоєння дисципліни **Rдис** (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи **Rнр** (до 70 балів): **R дис = R нр + R ат .**

8. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Презентації лекційного матеріалу та методичні вказівки до виконання лаобораторних робіт.

9. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Лавріщева К.І. Програмна інженерія. – К., 2008. – 319 с.
2. Ковалюк Т.В. Основи програмування. – С.-Пб.: Питер, 2005. – 384 с.

Додаткова

1. Соммервилл И. Инженерия програмного обеспечения. –6-е изд., пер. с англ. – М.: "Вильямс", 2002. – 624 с.

2. Липаев В.В. Программная инженерия. Методологические основы.– М.: ТЕИС, 2006.–608 с.

3. С. А. Орлов Программная инженерия. Учебник для вузов 5-е издание, обновленное и дополненное. Питер: 2003–640 с.

10. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Основи програмної інженерії (Електронний навчальний курс). Software Engineering Tutorial. –

2.Підручник "Програмна інженерія". – http://www.programsfactory.univ.kiev.ua/content/books/2