



КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
ФАКУЛЬТЕТ УПРАВЛІННЯ ФІНАНСАМИ ТА БІЗНЕСУ

ЗАТВЕРДЖЕНО
на засіданні кафедри економічної кібернетики
протокол № 1 від “27” серпня 2018 р.

В.о. зав. кафедри _____ Шевчук І.Б.
(підпис)

**ПЛАНИ ЛАБОРАТОРНИХ ЗАНЯТЬ
І МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ЩОДО ЇХ ПРОВЕДЕННЯ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Економіка і організація виробництва
програмних продуктів**

(назва навчальної дисципліни)

галузь знань: 05 «Соціальні та поведінкові науки»
(шифр та найменування галузі знань)

спеціальність: 051 “Економіка”
(код та найменування спеціальності)

спеціалізація: Інформаційні технології в бізнесі
(найменування спеціалізації)

освітній ступінь: бакалавр
(бакалавр/магістр)

Укладач:

Шевчук І.Б., доцент, к.е.н., доцент
(ПІБ, посада, науковий ступінь, вчене звання)

ЛЬВІВ 2018

1. МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ЩОДО ПРОВЕДЕННЯ ЛАБОРАТОРНИХ ЗАНЯТЬ

Лабораторна робота – це форма навчального заняття, при якому студент під керівництвом викладача особисто проводить натурні або імітаційні експерименти з метою практичного підтвердження окремих теоретичних положень навчальної дисципліни. При цьому він набуває навичок у роботі з лабораторним устаткуванням, обладнанням, обчислювальною технікою, вимірювальною апаратурою, методикою експериментальних досліджень у конкретній галузі.

Лабораторні роботи завжди виконуються після вивчення певного розділу (чи розділів) теоретичного курсу, оскільки вони є практичним підтвердженням теоретичних положень і висновків. Крім того, лабораторні заняття прищеплюють любов до предмета і сприяють розвитку навичок, які потрібні для практичної і винахідницької діяльності.

Лабораторні роботи відіграють важливу роль у підготовці фахівців через самостійну виконавчу діяльність. Виконуючи лабораторну роботу, студенти повинні проявляти ініціативу в подоланні труднощів. Звертатися до викладача лише тоді, коли вичерпано всі можливості вирішення питань по виконанню окремого етапу лабораторної роботи.

Дидактичні цілі проведення лабораторних робіт:

- опанування методами експериментальних досліджень (обробка результатів дослідів);
- формування відчуття спеціалізованої техніки та оволодіння специфікою практичної роботи;
- набуття навичок з безпеки проведення експериментальних досліджень;
- узагальнення, систематизація та поглиблення теоретичного матеріалу завдяки його практичному застосуванню;
- формування умінь застосовувати отримані знання в практичній діяльності, формування компетенцій;
- розвиток аналітичних та проєктувальних вмінь;
- вироблення самостійності, відповідальності і творчої ініціативи.

Отримані на понятійно-аналітичному рівні уміння та навички є суттєвим компонентом компетенції фахівця.

Тематика лабораторних робіт визначається робочою програмою дисципліни.

Лабораторне завдання (робота) може носити репродуктивний, частково-пошуковий і пошуковий характер. Роботи, що носять репродуктивний характер, відрізняються тим, що при їх проведенні студенти користуються детальними інструкціями, в яких вказані: мета роботи, пояснення (теорія, основні характеристики), порядок виконання роботи, таблиці, висновки (без формулювань), контрольні питання, література.

Роботи, що носять частково-пошуковий характер, відрізняються тим, що при їх проведенні студенти не користуються детальними інструкціями, їм не

заданий порядок виконання необхідних дій. В такому випадку студентам потрібно самим вибрати спосіб виконання лабораторної роботи.

Роботи, що носять пошуковий характер, відрізняються тим, що студенти повинні вирішити нову для них проблему, спираючись на наявні у них теоретичні знання.

Лабораторні заняття проводяться в спеціально обладнаних навчальних лабораторіях з використанням спеціального устаткування. В окремих випадках лабораторні заняття можуть проводитися в умовах реального професійного середовища.

Лабораторні роботи здебільшого належать до класу «жорстких», що виконуються згідно із заздалегідь підготовленим планом. Такі заняття мають на меті придбання та закріплення базових знань, відповідних умінь та навичок за відомим алгоритмом. Заняття передбачають активність студентів, але навчальні цілі обмежені відсутністю творчості студентів.

При організації підготовки до лабораторних робіт і їх виконання особливо увагу необхідно приділяти техніці безпеки.

Студент під час проведення лабораторних робіт повинен:

- беззаперечно дотримуватись правил охорони праці;
- ознайомитись з методичними рекомендаціями щодо проведення лабораторних робіт;
- виконати лабораторну роботу за відповідною методикою;
- скласти звіт про виконання лабораторної роботи;
- захистити результати лабораторної роботи;
- одержати оцінку за лабораторну роботу через визначену форму контролю.

Викладач повинен:

- провести інструктаж студентів щодо правил безпеки;
- керувати проведенням лабораторної роботи;
- здійснити поточний контроль опанування студентами методичних рекомендацій;
- забезпечити дотримання правил безпеки при виконанні лабораторної роботи;
- скласти графік консультацій;
- оцінити навчальну діяльність студента у процесі виконання лабораторної роботи.

У ході проведення лабораторного заняття викладач зобов'язаний забезпечити якісне виконання завдань, спрямувати студентів на творчий пошук.

Обов'язково за кожне лабораторне або практичне заняття викладач повинен оцінити рівень практичних умінь студента. Виконана робота реєструється в навчальному журналі. Усі студенти повинні зробити кожне завдання на лабораторну роботу з дисципліни.

2. КАЛЕНДАРНО-ТЕМАТИЧНИЙ ПЛАН ПРОВЕДЕННЯ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ заняття	Тема лабораторного заняття. Контрольні роботи (заліки по модулях)	Кількість
1	2	3
ЗАЛІКОВИЙ МОДУЛЬ № 1		
	Тема 2. Основні фактори, що визначають економічні характеристики виробництва програмних продуктів	2
1.	Лабораторна робота № 1	2
	Тема 3. Методи оцінки програмного продукту	4
2.	Лабораторна робота № 2	2
3.	Лабораторна робота № 2	2
	Тема 4. Основні метрики для оцінки програмного продукту	4
4.	Лабораторна робота № 3	2
5.	Лабораторна робота № 3	2
	Тема 5. Моделі оцінки вартості програмного забезпечення	2
6.	Лабораторна робота № 4	2
	Тема 6. Метрики об'єктно-орієнтованих програмних систем	4
7.	Лабораторна робота № 5	2
8.	Контрольна робота	2
	Разом лабораторних занять	14
	Разом контрольні роботи	2
	Разом годин	16

3. ПЛАНИ ЛАБОРАТОРНИХ ЗАНЯТЬ

План лабораторного заняття № 1

Тема. Розрахунок трудомісткості розробки програмного продукту.

Навчальний час: 2 год.

Міжпредметні зв'язки: Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху”, „Алгоритмізація та програмування”, “Об’єктно-орієнтоване програмування”, “Інтернет технології”, “Web-дизайн і Web-програмування”, “Технології проектування та адміністрування БД і СД”.

Мета і завдання лабораторного заняття: навчитися розраховувати трудомісткість програмного продукту.

Питання для перевірки базових знань за темою лабораторного заняття:

1. Які є класифікатори проекту?
2. У чому полягає суть методики А.Альбрехта?
3. Як визначити розмір коду прикладного програмного забезпечення?
4. Як визначити трудомісткість розробки ПЗ?
5. Що таке трудомісткість розробки?
6. Що таке бал функціональності?
7. Що таке логічний рядок коду?

Завдання:

1. Визначити функціональний розмір прикладного програмного забезпечення за нормативами значень класифікаторів проекту з табл. 1.1 за варіантами, які обираються відповідно до порядкового номера в журналі академічної групи:

Варіант	ПЗ	Варіант	ПЗ
1	Готове програмне забезпечення, яке потребує налаштування, для автоматизації бізнес процесу одного структурного підрозділу, замовлене місцевим виконавчим органом	9	База даних для автоматизації бізнес-процесів одного відомства з територіальними підрозділами, замовлена державним органом, діяльність якого пов'язана з безпекою для життя
2	База даних для автоматизації бізнес-процесів одного відомства, замовлена центральним державним органом	10	Клієнт-серверна система (тонкий клієнт) для автоматизації бізнес-процесів декількох відомств. Замовлена місцевим виконавчим органом
3	Клієнт-серверна система(товстий клієнт) для автоматизації бізнес-процесів одного відомства з територіальними підрозділами, замовлена державним органом,	11	Клієнт-серверна система (товстий клієнт) для автоматизації бізнес-процесів декількох відомств і інтеграція з зовнішніми

	діяльність якого пов'язана з небезпекою для життя		інформаційними системами, замовлена центральним державним органом
4	Сервіс-орієнтована система для автоматизації бізнес-процесів декількох відомств, замовлена місцевим виконавчим органом	12	Готове програмне забезпечення, яке потребує налаштування, для автоматизації бізнес-процесів декількох відомств і інтеграція з зовнішніми інформаційними системами, замовлене державним органом, діяльність якого пов'язана з небезпекою для життя
5	Клієнт-серверна система (тонкий клієнт) для автоматизації бізнес-процесів декількох відомств і інтеграція з зовнішніми інформаційними системами, замовлена центральним державним органом	13	База даних для автоматизації бізнес-процесів декількох відомств, замовлена місцевим виконавчим органом
6	Сервіс-орієнтована система для автоматизації бізнес-процесів відомства і інтеграція з зовнішніми інформаційними системами, замовлена державним органом, діяльність якого пов'язана з небезпекою для життя	14	Сервіс-орієнтована система для автоматизації бізнес процесу одного структурного підрозділу, замовлена центральним державним органом
7	Клієнт-серверна система (товстий клієнт) для автоматизації бізнес процесу одного структурного підрозділу, замовлена місцевим виконавчим органом	15	Клієнт-серверна система (тонкий клієнт), для автоматизації бізнес-процесів одного відомства, замовлена державним органом, діяльність якого пов'язана з небезпекою для життя
8	Готове програмне забезпечення, яке потребує налаштування, для автоматизації бізнес-процесів одного відомства, замовлене центральним державним органом	16	База даних для автоматизації бізнес-процесів відомства і інтеграція з зовнішніми інформаційними системами, замовлена місцевим виконавчим органом

Таблиця 1.1

Нормативи значень класифікаторів проекту

К₁ – масштаб об'єкту автоматизації	К₂ – тип замовника	К₃ – тип програмного забезпечення
Автоматизація бізнес процесу одного структурного підрозділу 1	Місцевий виконавчий орган- 8	Готове програмне забезпечення, яке потребує налаштування- 1

Автоматизація бізнес-процесів одного відомства- 8	Центральний державний орган- 14	База даних- 6
Автоматизація бізнес-процесів одного відомства з територіальними підрозділами- 9	Державний орган, діяльність якого пов'язана з небезпекою для життя- 15	Клієнт-серверна система (товстий клієнт)- 8
Автоматизація бізнес-процесів відомства інтеграція з зовнішніми інформаційними системами - 10		Клієнт-серверна система (тонкий клієнт)- 11
Автоматизація бізнес-процесів декількох відомств- 12		Сервіс-орієнтована система 15
Автоматизація бізнес-процесів декількох відомств інтеграція з зовнішніми інформаційними системами - 13		

2. Визначити розмір коду прикладного програмного забезпечення, написаного мовою програмування із табл. 1.2 за варіантами, які обираються відповідно до порядкового номера в журналі академічної групи:

Варіант	Мова програмування	Варіант	Мова програмування
1	1, 17, 33, 49, 66, 82	9	9, 28, 42, 58, 71, 85
2	2, 18, 39, 56, 73, 81	10	10, 24, 36, 55, 75, 89
3	3, 19, 43, 60, 76, 87	11	11, 32, 45, 63, 78, 93
4	4, 20, 34, 54, 70, 90	12	12, 29, 47, 62, 68, 96
5	5, 21, 40, 61, 79, 83	13	13, 25, 37, 52, 80, 84
6	6, 22, 35, 50, 65, 95	14	14, 31, 48, 64, 72, 92
7	7, 27, 46, 57, 74, 86	15	15, 30, 44, 59, 77, 88
8	8, 23, 41, 51, 67, 91	16	16, 26, 38, 53, 69, 94

Таблиця 1.2
Коефіцієнти переводу балів функціональності в кількість логічних рядків коду

№	Мова програмування	КП (кількість логічних рядків коду на одну функціональну точку)
	1	2
2.	BasicAssembler	320
3.	Autocoder	320
4.	Netron/CAP	296
5.	MacroAssembler	213

6.	C	128
7.	Пакетні файли DOS	128
8.	Basic	107
9.	Макроси Lotus	107
10.	ALGOL	105
11.	COBOL	105
12.	FORTRAN	105
13.	JOVIAL	105
14.	Змішані мови програмування	105
15.	JCL	96
16.	VPF	95
17.	Pascal	91
18.	COBOL(ANSI85)	91
19.	APS	86
20.	Slogan	81
21.	RPG	80
22.	Modula-2	80
23.	PL/1	80
24.	Паралельний Pascal	80
25.	Fortran95	71
26.	Mantis	71
27.	Sabretalk	70
28.	Mapper	69
29.	ColdFusion	68
30.	Datastage	67
31.	Ideal	66
32.	Basic (ANSI)	64
33.	FORTH	64
34.	LISP	64
35.	PROLOG	64
36.	Powerhouse	63
37.	Uniface	61
38.	.NET	60
39.	JSP	59
40.	LOGO	58
41.	C#	58
42.	J2EE	57
43.	Розширений LISP	56
44.	RPGIII	56
45.	ASP	56
46.	Java	55
47.	JavaScript	54
48.	C++	53
49.	YACC	53
50.	Culprit	51
51.	Natural	51
52.	KML	50
53.	VisualBasic	50
54.	REXX	50
55.	Ada95	49

56.	PL/SQL	47
57.	CICS	46
58.	SIMULA	46
59.	Taskmate	45
60.	Focus	45
61.	WebScripts	44
62.	Parabase	42
63.	Мовибазданих	40
64.	ClipperDB иdBase III	40
65.	Informix	40
66.	OracleиSYBASE	40
67.	Openroad	39
68.	Access	38
69.	VBScript	38
70.	Advantage	38
71.	PeopleSoft	37
72.	Cool:Gen/IEF	37
73.	DBase IV	36
74.	FoxPro2.5	34
75.	APL	32
76.	Статичнімови(SAS)	32
77.	Maestro	30
78.	DELPHI	29
79.	Стандартні об'єктно-орієнтовані мови	29
80.	Powerbuilder	28
81.	VB.Net	28
82.	OBJECTIVE-C	27
83.	Lotus Script	23
84.	OracleDeveloper /2000	23
85.	Smalltalk	21
86.	awk	21
87.	EIFFEL	21
88.	Shell-сценарії(Perl)	21
89.	Стандартні мови 4-го покоління (4GL)	20
90.	OR3(4GL)	20
91.	ApplicationBuilder	20
92.	CORBA	20
93.	CristalReports	20
94.	Datatrieve	20
95.	CLIPPER	19
96.	ABAP(SAP)	18
97.	HTML 3.0	15
98.	SiebelTools	13
99.	SQL	13
100.	Easytrieve+	13
101.	SQLForms	11
102.	Excel	6
103.	QUATTROPRO	6
104.	Мови створенняпiктограм	4

3. Визначити трудомісткість розробки прикладного програмного забезпечення (за даними табл. 1.3-1.4).

Таблиця 1.3

Показники розробки за рівнями

Показники розробки, R _j	Низький рівень	Середній рівень	Високий рівень
передбачуваність проекту для розроблювача, R ₁	4,96	3,72	2,48
Гнучкість процесу розробки, R ₂	4,05	3,04	2,03
Ступінь знищення ризиків, R ₃	5,65	4,24	2,83
згуртованість команди проекту, R ₄	4,38	3,29	2,19
Зрілість процесів в організації розроблювача, R ₅	6,24	4,68	3,12

Таблиця 1.4

Множники витрат

Множники витрат, Z _i	Низький рівень	Середній рівень	Високий рівень
Згуртованість персоналу, Z ₁	1,20	1,00	0,83
надійність і складність прикладного програмного забезпечення, Z ₂	0,83	1,00	1,33
Складність платформи, Z ₃	0,87	1,00	1,29
необхідний рівень повторного використання, Z ₄	0,95	1,00	1,07
Досвідченість персоналу, Z ₅	1,22	1,00	0,87
Використання інструментів, Z ₆	1,10	1,00	0,87
Щільність графіка проекту, Z ₇	1,14	1,00	1,00

Форми контролю знань – презентація виконаних завдань, обговорення виконаних завдань.

Рекомендована література до теми лабораторного заняття:

Основна та допоміжна література:

1. Орлов С.А. Технологии разработки программного обеспечения. – СПб.: Питер, 2002. – 464 с.
2. Липаев В.В. Экономика производства сложных программных продуктов. – М.: Синтег, 2008. – 432 с.
3. Изосимов А.В., Рыжко А.Л. Метрическая оценка качества программ. – М.: МАИ. – 1989. – 450 с.
4. Методика расчетов трудоемкости и стоимости работ на разработку информационных систем. – АО «НИТ». – 20 с.

Інтернет ресурси:

1. Колдовский В. Разработка ПО: метрики программных проектов. [Електронний ресурс]. http://itc.ua/articles/razrabotka_po_metriki_programmnyh_proektov_27774
2. Лекции по управлению программными проектами [Електронний ресурс]. http://citforum.ru/SE/project/arkhipenkov_lectures/

Обладнання заняття, ТЗН тощо:ноутбук, ПЕОМ.

Завдання студентам на самостійне опрацювання навчального матеріалу, рішення задач, розв'язання вправ для підготовки до наступного лабораторного заняття.

1. Вивчити, що являє собою нормативна трудомісткість розробки ПЗ.
2. Вияснити, яка різниця між нормативною і загальною трудомісткістю розробки ПЗ.

Укладач: _____ Шевчук І.Б., доцент, к.е.н., доцент
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)

План лабораторного заняття № 2

Тема. Метрики оцінки складності програмного продукту.

Навчальний час: 4 год.

Міжпредметні зв'язки: Зв'язок із елементами знань і умінь таких навчальних дисциплін як „Вступ до фаху”, „Алгоритмізація та програмування”, “Об’єктно-орієнтоване програмування”, “Інтернет технології”, “Web-дизайн і Web-програмування”, “Технології проектування та адміністрування БД і СД”.

Мета і завдання лабораторного заняття: навчитися розраховувати метричні оцінки Холстеда, цикломатичної складності Мак-Кейба та Чепіна.

Питання для перевірки базових знань за темою лабораторного заняття:

1. Що таке критерій якості? Перерахуйте його основні характеристики.
2. Що таке метрика якості програми?
3. На які три групи поділяються метрики для оцінки якості ПЗ?
4. Які метричні шкали існують?
5. Які основні групи метрик виділяють при оцінці складності програм?
6. Перерахуйте метричні характеристики, що використовуються в метриках Холстеда?
7. Яким чином розраховується метрика довжини програми?
8. Яким чином визначається метрика обсягу програми?
9. Чим метрика потенційного обсягу відрізняється від метрики обсягу?
10. Як встановити рівень реалізації програми?
11. Як визначається інтелектуальний зміст конкретного алгоритму?
12. Яким чином розраховується оцінка необхідних інтелектуальних зусиль з написання програми?
13. Опишіть методику знаходження цикломатичного числа Мак-Кейба.
14. Від чого залежить цикломатичне число Мак-Кейба і що воно не враховує?
15. Опишіть методику знаходження метрики Джілбі.
16. Які метрики відносяться до метрик складності потоку даних?
17. Що показує метрика звернення до глобальних змінних?
18. Як визначається метрика Спен?
19. Для чого використовується метрика Чепіна і яким чином вона розраховується?
20. Як знайти рівень коментування програми?
21. Яким чином розподіляються коментарі в програмі?
22. Чим відрізняється метрика та міра якості?

Завдання:

1. Розрахувати метрики Холстеда для програми написаної мовою С або іншою мовою програмування (програмний код взяти власний або з мережі Інтернет).

1.1. Програмний код оформити наступним чином:

Номер рядки	Рядки програми
1.	using System;
2.	namespace Ex
3.	{
4.	class Program
5.	{
6.	static void Main()
7.	{
8.	double x,y,F;
9.	char check;
10.	do
11.	{
12.	Console.WriteLine("Введіть значення змінної x");
13.	Console.Write("x=");
14.	x=double.Parse(Console.ReadLine());
15.	Console.WriteLine("Введіть значення змінної y");
16.	Console.Write("y=");
17.	y=double.Parse(Console.ReadLine());
18.	if (x<y)
19.	F=Math.Sin(x)+Math.Cos(y)*Math.Cos(y);
20.	else
21.	if (x==y)
22.	F=Math.Log(x);
23.	else
24.	F=Math.Sin(x)*Math.Sin(x)+Math.Cos(y);
25.	Console.WriteLine("F="+F);
26.	Console.WriteLine("Запустити програму знову? Y/N");
27.	check=char.Parse(Console.ReadLine());
28.	}while (check=='Y' check=='y');
29.	}
30.	}
31.	}

1.2. Використовуючи метрику Холстеда, визначити:

- 1) n1 - кількість різних операторів даної реалізації;
- 2) n2 - кількість різних операндів даної реалізації;

- 3) N_1 – загальну кількість всіх операторів;
 4) N_2 – загальну кількість всіх операндів;
 5) n_2^* - кількість різних вхідних і вихідних операндів.
 6) на основі наведених вище характеристик обчислити:
 n – довідник,
 N - довжину програми,
 \tilde{N} – (оціночну) довжину програми,
 V – обсяг програми,
 V^* - потенціальний (мінімальний) обсяг,
 D - складність програми,
 L – рівень програми,
 I – інтелектуальний вміст програми,
 E – роботу з програмування,
 T' - наближений час програмування,
 A - рівень мови,
 B – рівень помилок.

1.3. Заповнити наступні таблиці:

Таблиця 2.1

Довідник операторів і операцій програми

№з/п	Оператори, операції	Номера рядків	Кількість повторів

Таблиця 2.2

Довідник операндів

№з/п	Операнди	Номери рядків	Кількість повторів

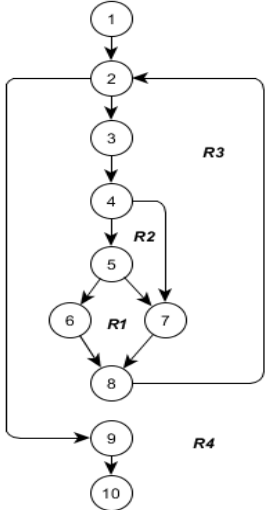
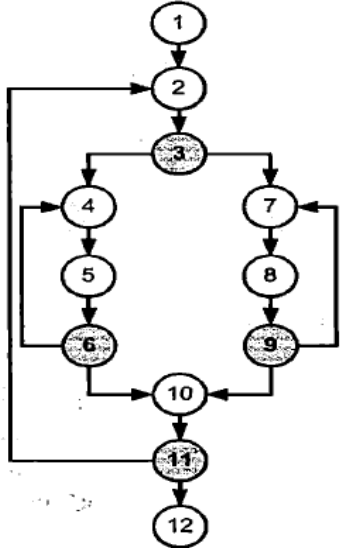
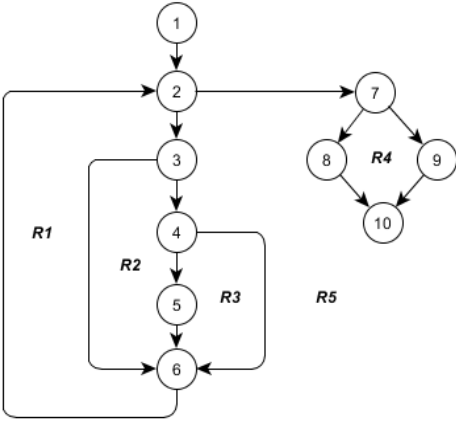
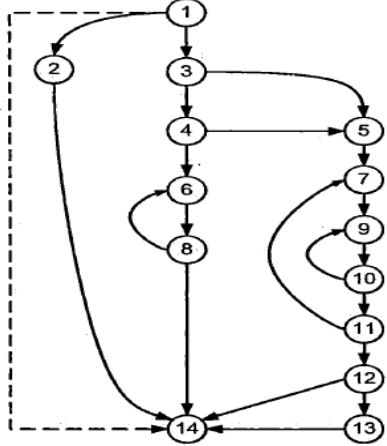
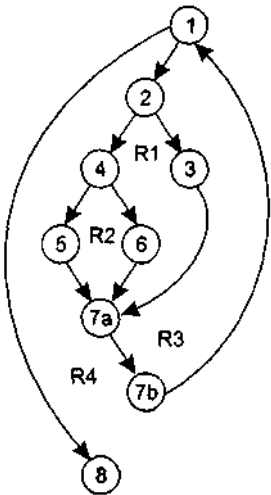
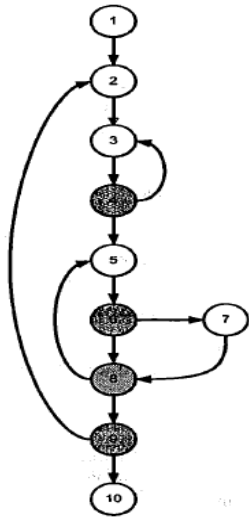
Таблиця 2.3

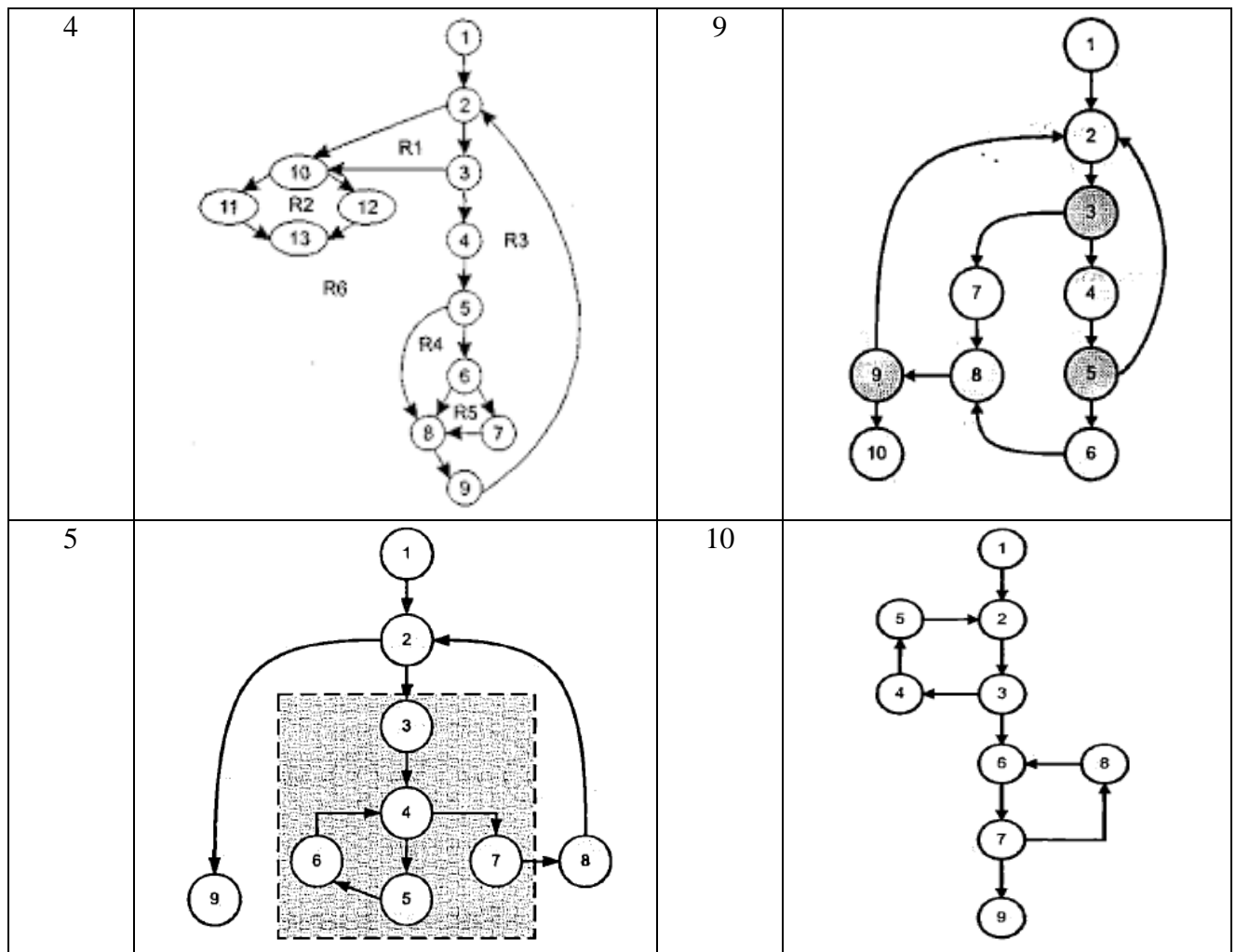
Вхідні і вихідні змінні

Вхідні змінні	Вихідні змінні

2. Для використаного програмного коду у завданні 1 розрахувати метрику Чепіна.

3. За допомогою метрики Мак-Кейба побудувати граф управління програмою. Знайти цикломатичну складність графа і цикломатичне число за варіантом.

Варіант		Варіант	
1		6	
2		7	
3		8	



Форми контролю знань – презентація виконаних завдань, обговорення виконаних завдань.

Рекомендована література до теми лабораторного заняття:

Основна та допоміжна література:

1. Орлов С.А. Технологии разработки программного обеспечения. – СПб.: Питер, 2002. – 464 с.
2. Липаев В.В. Экономика производства сложных программных продуктов. – М.: Синтег, 2008. – 432 с.
3. Изосимов А.В., Рыжко А.Л. Метрическая оценка качества программ. – М.: МАИ. – 1989. – 450 с.
4. Методика расчетов трудоемкости и стоимости работ на разработку информационных систем. – АО «НИТ». – 20 с.

Інтернет ресурси:

1. Колдовский В. Разработка ПО: метрики программных проектов. [Електронний ресурс]. http://itc.ua/articles/razrabotka_po_metriki_programmnyh_proektov_27774

2. Лекции по управлению программными проектами [Электронный ресурс].
http://citforum.ru/SE/project/arkhipenkov_lectures/

Обладнання заняття, ТЗН тощо: ноутбук, ПЕОМ.

Завдання студентам на самостійне опрацювання навчального матеріалу, рішення задач, розв'язання вправ для підготовки до наступного лабораторного заняття.

1. Ознайомитись із основними положеннями методів проведення експертних оцінок.
2. Практичне застосування методу групової оцінки трудозатрат Wideband Delphi.
3. Особливості проведення експертного оцінювання ПП.

Укладач: _____ Шевчук І.Б., доцент, к.е.н., доцент
(підпис) (ПІБ, посада, науковий ступінь, вчене звання)

[Продовжити перегляд](#)

4. КРИТЕРІЇ ОЦІНЮВАННЯ

Оцінювання рівня знань студентів на лабораторних заняттях проводиться за 5-ти бальною шкалою (від 1 до 5 балів).

Оцінки за виконання лабораторних робіт враховуються як результат поточного контролю знань студента.

Порядок вивчення та оцінювання доводиться до відома студентів протягом семестру.

Система нарахування рейтингових балів та критерії оцінювання знань студентів

№ з/п	Види робіт. Критерії оцінювання знань студентів	Бали рейтингу	Максимальна кількість балів
1. Бали поточної успішності за участь у лабораторних заняттях			
Критерії оцінювання		5 балів	
лабораторна робота виконана у зазначений термін, у повному обсязі, без помилок		5	
лабораторна робота виконана у зазначений термін, у повному обсязі, але є незначні помилки		4	
лабораторна робота виконана у неповному обсязі, або (та) з порушенням терміну її виконання, або (та) при наявності значних помилок		3	
виконання пропущеної без поважних причин лабораторної роботи або повторне виконання незарахованої лабораторної роботи		2	
лабораторна робота не виконана або не зарахована		0-1	