



МІЖНАРОДНИЙ ЄВРОПЕЙСЬКИЙ УНІВЕРСИТЕТ

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ЄВРОПЕЙСЬКА ШКОЛА БІЗНЕСУ»**

ЗАТВЕРДЖЕНО

Директор ННІ

«Європейська школа бізнесу»


Юлія РЕМИГА

від «11» 09 2023 р.

М.П.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ**

Рівень вищої освіти

перший (бакалаврський)

(назва)

Спеціальність

121 Інженерія програмного забезпечення

(код і назва спеціальності)

Освітня програма

«Інженерія програмного забезпечення»

(назва освітньої програми)

Київ – 2023

Робоча програма навчальної дисципліни «Основи програмної інженерії» складена на основі освітньо-професійної програми 121 «Інженерія програмного забезпечення» для першого (бакалаврського) рівня спеціальності 121 «Інженерія програмного забезпечення», затвердженої Вченою радою Університету «30» травня 2023 року, протокол № 4.

Укладач: Нестеренко Олександр Васильович, доктор технічних наук, доцент

Рецензент: Фаловський Олександр Олександрович, к.т.н.

Гарант освітньої програми:  Олександр НЕСТЕРЕНКО, д.т.н., доцент

Робочу програму навчальної дисципліни розглянуто та схвалено кафедрою інформаційних технологій, протокол № 1 від 31.08.2023 р.

Розглянуто і схвалено Вченою радою Навчально-наукового інституту «Європейська школа бізнесу», протокол № 1 від «11» вересня 2023 р.

ВСТУП

Програма вивчення навчальної дисципліни «Основи програмної інженерії» складена відповідно до Стандарту вищої освіти України (далі – Стандарт) галузі знань 12 «Інформаційні технології» спеціальності 121 «Інженерія програмного забезпечення».

Опис навчальної дисципліни (анотація). Дана навчальна дисципліна є однією з дисциплін фундаментальної природничо-наукової підготовки майбутніх розробників програмного забезпечення.

Таблиця 1

Найменування показників	Галузь знань, напрям підготовки, освітній рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 4	Галузь знань, 12 «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»	Нормативна	
Розділів – 2	Спеціальність: 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»	Рік підготовки	
Змістових розділів – 4		2023-24	2023-24
Індивідуальне науково-дослідне завдання: стартап		Семестр	
		2	2
Загальна кількість годин – 120		Лекції	
		16	6
Тижневе навантаження: аудиторних – 3 самостійної роботи студента – 4	Практично-лабораторні		
	32	4	
	Самостійна робота		
	72	110	
	Освітній рівень: бакалавр	Вид контролю:	
		залік	залік

Предметом вивчення навчальної дисципліни є основні поняття програмної інженерії.

Міждисциплінарні зв'язки: Навчальна дисципліна є вступом до спеціальності і базисом вивчення багатьох дисциплін з подальшого навчання.

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1.1. **Метою** викладання навчальної дисципліни «Основи програмної інженерії» є оволодіння студентами знаннями про принципи та процеси розробки програмних продуктів та програмних систем, сучасні методології та практики у розробці програмного забезпечення, набуття умінь і практичних навичок щодо використання основних засобів організації процесу розробки програмного забезпечення.

1.2. Основними **завданнями** вивчення дисципліни «Основи програмної інженерії» є:

- ознайомити студентів з основними принципами та концепціями програмної інженерії;
- розвинути вміння аналізувати, проектувати та реалізовувати програмне забезпечення;
- вивчити основні методи та інструменти, що використовуються в програмній інженерії;
- підготувати студентів до подальшого вивчення спеціалізованих курсів з програмування та розробки програмного забезпечення.

1.3. **Компетентності та результати навчання**, формуванню яких сприяє дисципліна (взаємозв'язок з нормативним змістом підготовки здобувачів вищої освіти, сформульованим у термінах результатів навчання у Стандарті).

Згідно з вимогами стандарту дисципліна забезпечує набуття студентами **компетентностей**:

Таблиця 2

Інтегральна компетентність	Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій.
Загальні компетентності	ЗК 02. Здатність застосовувати знання у практичних ситуаціях. ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
Спеціальні (фахові, предметні) компетентності	СК5. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу програмного забезпечення. СК8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення. СК9. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності. СК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя. СК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

Інтегративні кінцеві програмні результати навчання, формуванню яких сприяє навчальна дисципліна:

Програмні результати навчання	<p>ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. ПР02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності. ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення. ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення. ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення. ПР16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації. ПР22. Знати та вміти застосовувати методи та засоби управління проектами.</p>
--------------------------------------	--

Після опанування дисципліни студент повинен

знати:

- поняття інженерії програмного забезпечення;
- особливості створення програмних продуктів і систем;
- основні етапи розробки програмних продуктів і систем;
- основні методології розробки програмного забезпечення та їх застосування;
- сучасні практики, що застосовуються у процесі розробки програмного забезпечення;

вміти:

- раціонально визначати та обґрунтовувати необхідну методологію розробки програмного забезпечення та модель життєвого циклу в залежності від його властивостей та особливостей команди розробки;
- створювати детальну функціональну специфікацію програмного продукту на основі наданих замовником вимог;
- розробляти плани проекту розробки програмного забезпечення;
- застосовувати технології роботи з спеціальними програмними застосуваннями, здійснювання обробки даних та їх аналізу.

2. ІНФОРМАЦІЙНИЙ ОБСЯГ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

На вивчення навчальної дисципліни «Основи програмної інженерії» відводиться 120 годин 4 кредити ЄКТС.

ЗМІСТОВИЙ РОЗДІЛ 1

ВИЗНАЧЕННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

Тема 1.1. Загальне визначення програмної інженерії.

Тема 1.2. Області знань SWEBOOK.

Тема 1.3. Стандарти і моделі життєвого циклу.

ЗМІСТОВИЙ РОЗДІЛ 2 ВИМОГИ ДО ПРОГРАМНИХ СИСТЕМ

Тема 2.1. Інженерія вимог.

Тема 2.2. Моделювання та проектування архітектури програмних систем.

Тема 2.3. Методи програмування

ЗМІСТОВНИЙ РОЗДІЛ 3 ІНЖЕНЕРІЯ ВИРОБНИЦТВА ПРОГРАМНИХ ПРОДУКТІВ

Тема 3.1. Інженерія виробництва.

Тема 3.2. Методи доведення, верифікації і тестування програм.

Тема 3.3. Моделі якості та надійності програмних систем.

Тема 3.4. Управління програмним проектом.

3. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви розділів і тем							Кількість годин				
							Всього	у тому числі			СРС
								Лекції	Практичні/ семінари	Лабораторні	
1	2	3	4	5	6	1	Розділ 1. Змістовний розділ 1. ВИЗНАЧЕННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ				
Розділ 1.											
Змістовний розділ 1. ВИЗНАЧЕННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ											
<i>Тема 1.1.</i> Загальне визначення програмної інженерії							7	1			6
<i>Тема 1.2.</i> Области знань SWEBOOK							9	1		2	6
<i>Тема 1.3.</i> Стандарти і моделі життєвого циклу							12	2		4	6
Разом за розділом 1							28	4		6	18
Змістовний розділ 2. ВИМОГИ ДО ПРОГРАМНИХ СИСТЕМ											
<i>Тема 2.1.</i> Інженерія вимог							9	1		2	6
<i>Тема 2.2.</i> Моделювання та проектування архітектури програмних систем							11	1		4	6
<i>Тема 2.3.</i> Методи програмування							12	2		4	6
Разом за розділом 2							32	4		10	18
Розділ 2.											
Змістовний розділ 3. ІНЖЕНЕРІЯ ВИРОБНИЦТВА ПРОГРАМНИХ ПРОДУКТІВ											
<i>Тема 3.1.</i> Інженерія виробництва							15	2		4	9
<i>Тема 3.2.</i> Методи доведення, верифікації і тестування програм							15	2		4	9
<i>Тема 3.3.</i> Моделі якості та надійності програмних систем							15	2		4	9
<i>Тема 3.4.</i> Управління програмним проектом							15	2		4	9
Разом за розділом 3							60	8		16	36
<i>Разом</i>							120	16		32	72
<i>Консультації</i>											
<i>Екзамен</i>											
Всього годин							120				

4. ТЕМИ ЛЕКЦІЙ

№ лекції	Назва теми лекції та перелік основних питань
1	<p>ЗАГАЛЬНЕ ВИЗНАЧЕННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ. Роль програмної інженерії в розробці програмного забезпечення. Програмна інженерія як інженерна та наукова дисципліна. Процес розробки програмного забезпечення. Ознайомлення з основними поняттями та термінологією програмної інженерії. Основні міжнародні рекомендації з ПІ. Коротка історична довідка розвитку ПІ. Огляд сучасних тенденцій у програмній інженерії. Інновації та перспективи розвитку програмного забезпечення. Структура дисципліни та організація її вивчення.</p> <p>ОБЛАСТІ ЗНАНЬ SWEBOOK. Характеристика областей знань з інженерії програмного забезпечення. Інженерія вимог. Проектування програмного забезпечення. Конструювання програмного забезпечення. Тестування програмного забезпечення. Супровід програмного забезпечення. Керування конфігурацією. Керування інженерією програмного забезпечення. Базовий процес ПІ. Методи і інструменти ПІ. Якість програмного забезпечення.</p>
2	<p>СТАНДАРТИ І МОДЕЛІ ЖИТТЄВОГО ЦИКЛУ. Характеристика життєвого циклу за стандартом ISO/IEC 12207. Прикладні моделі життєвого циклу. Типи моделей життєвого циклу (каскадна, інкрементна, спіральна, еволюційна та ін.)</p>
3	<p>ІНЖЕНЕРІЯ ВИМОГ. Визначення та аналіз вимог до програмного забезпечення. Загальні підходи до визначення вимог. Класифікація вимог. Вимоги до функціональності, надійності, продуктивності та інші характеристики програмного забезпечення. Інженерія вимог. Методи інженерії вимог: ідентифікація, документування, аналіз та верифікація вимог. Аналіз і збирання вимог.. Фіксація вимог. Трасування вимог.</p> <p>МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНИХ СИСТЕМ. Принципи та методи проектування програмного забезпечення. Моделювання систем та компонентів програмного забезпечення. Основні поняття об'єктно-орієнтованих методів аналізу. Метод побудови об'єктної моделі предметної області. Проектування архітектури програмних систем. Архітектурні шаблони та стилі. Загальні підходи до проектування програмних систем. Проектування різних видів архітектур програмних систем</p>
4	<p>МЕТОДИ ПРОГРАМУВАННЯ. Прикладне (систематичне) програмування (структурний метод; об'єктно-орієнтований; компонентний; сервісно-орієнтоване; аспектно-орієнтований та ін.).</p>
5	<p>ІНЖЕНЕРІЯ ВИРОБНИЦТВА. Інженерія компонентів повторного використання. Прикладна інженерія та інженерія предметної області. Інженерія індустріального виробництва програмних продуктів. Оцінювання вартості системи з компонентів</p>
6	<p>МЕТОДИ ДОВЕДЕННЯ, ВЕРИФІКАЦІЇ І ТЕСТУВАННЯ ПРОГРАМ. Мови специфікації програм і їхня класифікація. Методи доведення правильності програм. Верифікація і валідація програм. Тестування програмних систем. Інфраструктура перевірки правильності програмних систем.</p>
7	<p>МОДЕЛІ ЯКОСТІ ТА НАДІЙНОСТІ ПРОГРАМНИХ СИСТЕМ. Модель якості програмних систем. Стандартні показники якості та метрики якості. Ґрунтовні поняття проблематики надійності. Моделі оцінки надійності програмних систем. Сертифікація програмного продукту.</p>
8	<p>УПРАВЛІННЯ ПРОГРАМНИМ ПРОЕКТОМ. Менеджмент проекту. РМВОК. Галузі знань з управління проектами. Модель проектного менеджменту.</p>

	Інфраструктура програмного проекту. Розподіл робіт і ролей у проекті. Методи планування і керування проектом (графічні, планування і контроль, оцінювання вартості проекту). Керування ризиками у проекті. Керування конфігурацією системи. Інструмент Microsoft Project.
--	---

5. ТЕМИ ЛАБОРАТОРНО-ПРАКТИЧНИХ ЗАНЯТЬ

№ з/п	Назва лабораторної роботи (комп'ютерного практикуму)	Кількість годин
1	Області знань SWEBOOK	2
2	Стандарти і моделі життєвого циклу	4
3	Інженерія вимог	2
4	Знайомство з продуктовим ІТ	4
5	Засади створення та розвитку ІТ-продуктів	4
6	Спільна робота над проектом продукту. Модель Lean Canvas	4
7	Пошук ідею продукту та програмного рішення	4
8	Аналіз поведінки споживача	4
9	Створення пітчдеків стартапів для залучення інвесторів	6
	Всього	32

6. САМОСТІЙНА РОБОТА

№ з/п	Назва теми, що виноситься на самостійне опрацювання	Кількість годин
Змістовний розділ 1. ВИЗНАЧЕННЯ ПРОГРАМНОЇ ІНЖЕНЕРІЇ		
1	<i>Тема 1.1.</i> Реферат щодо історичного шляху розвитку програмування.	6
2	<i>Тема 1.2.</i> Ознайомитись з матеріалами рекомендацій SE2004 та CC2020	6
3	<i>Тема 1.3.</i> Ознайомитись з матеріалом Selecting a development approach	6
Змістовний розділ 2. ВИМОГИ ДО ПРОГРАМНИХ СИСТЕМ		
3	<i>Тема 2.1.</i> Вимоги до характеристик програмного забезпечення	6
4	<i>Тема 2.2.</i> Метод побудови об'єктної моделі предметної області.	6
5	<i>Тема 2.3.</i> Теоретичне програмування	6
Змістовний розділ 3. ІНЖЕНЕРІЯ ВИРОБНИЦТВА ПРОГРАМНИХ ПРОДУКТІВ		
6	<i>Тема 3.1.</i> Описати існуючий продукт в обраній сфері за моделлю Lean Canvas	9
7	<i>Тема 3.2.</i> Підготувати план запуску власного продукту	9
8	<i>Тема 3.3.</i> Проаналізувати будь-який цікавий додаток на предмет наявності в ньому законів та принципів з UX-дизайну	9
9	<i>Тема 3.4.</i> Підготувати презентацію свого пітчдеку за визначеною структурою	9
		72

7. МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни «Основи програмної інженерії» застосовуються інформаційні та практичні методи навчання: класичні лекції, лекції-дискусії та лабораторно-практичні заняття з використанням

симуляційних лабораторних практикумів, а також консультації з виконання самостійної роботи студентів, письмові завдання при проведенні контрольних робіт.

Особливістю навчання є використання вбудованого курсу «Створення та розвиток ІТ-продуктів» від компанії Genesis з метою стажування студентів в ІТ-бізнесі та пришвидшення розробки ними проектів стартапів.

Методи навчально-пізнавальної діяльності: пояснювально-ілюстративний метод, репродуктивний метод, метод проблемного викладу, частково-пошуковий або евристичний метод, дослідницький метод.

Методи стимулювання й мотивації навчально-пізнавальної діяльності: індуктивні і дедуктивні методи навчання, методи стимулювання і мотивації навчання.

8. МЕТОДИ КОНТРОЛЮ

Відповідно до плану вивчення дисципліни «Основи програмної інженерії» передбачається проведення поточного та підсумкового контролю.

Поточний контроль – оцінювання рівня знань, умінь та навичок осіб, які навчаються, що здійснюється в ході навчального процесу шляхом проведення тестування по закінченню лекцій. Додатково враховується результати виконання лабораторних робіт, а також сертифікаційного оцінювання за проходження студентами стажування на платформі курсу «Створення та розвиток ІТ-продуктів».

9. ФОРМА ПІДСУМКОВОГО КОНТРОЛЮ УСПІШНОСТІ НАВЧАННЯ

Формою підсумкового контролю є залік, який складається очно (при особливій ситуації – у формі комп'ютерного тесту) в період призначений деканатом або за індивідуальним графіком, який затверджується навчальним планом.

10. СХЕМА НАРАХУВАННЯ ТА РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ

Накопичення балів протягом семестру

№ з/п	Вид діяльності	Кількість балів за дидактичну одиницю	Кількість	Загальна кількість балів
1	Тестування на лекціях	2	8	16
2	Виконання лабораторних робіт	2-4	9	18-36
3	Оцінка за сертифікатом стажування	30-60	1	30-60
4	Залік			10-20
Максимальна оцінка				100

Загальна оцінка знань студентів за поточним контролем

Результати поточного контролю знань студентів в цілому оцінюються в діапазоні від **0** до **60** балів.

Студент допускається до підсумкового контролю за умови виконання вимог навчальної програми та у разі, якщо за поточну навчальну діяльність він набрав не менше **36** балів.

Підсумкове оцінювання знань студентів

Підсумкове оцінювання знань студентів проводиться у формі заліку.

Розподіл балів оцінювання при підсумковому контролі з навчальної дисципліни

цінка в балах за поточне оцінювання	Оцінка за національною шкалою
90-100	Відмінно
70-89	Добре
60-69	Задовільно
менше 60	Незадовільно

Під час оцінювання відповіді на окреме питання під час екзамену додатково враховуються допущені недоліки та помилки, якими вважаються:

– неохайне оформлення роботи (не загальноприйняті скорочення, незрозумілий почерк, використання олівців замість чітких чорнил) (мінус **2** бали);

– неточності в назвах окремих термінів та понять (мінус **4** бали).

Критерії оцінювання відповіді на теоретичні питання:

1. Повна відповідь на питання, яка оцінюється **«відмінно»**, повинна відповідати таким вимогам:

– розгорнутий, вичерпний виклад змісту даної у питанні проблеми;
– повний перелік необхідних для розкриття змісту питання фізичних термінів та законів;

– здатність здійснювати порівняльний аналіз різних теорій, концепцій, підходів та самостійно робити логічні висновки й узагальнення;

– уміння користуватись методами наукового аналізу фізичних явищ, процесів і характеризувати їхні риси та форми виявлення;

– демонстрація здатності висловлення та аргументування власного ставлення до альтернативних поглядів на дане питання;

2. Відповідь на питання оцінюється **«добре»**, якщо:

– відносно відповіді на найвищий бал не зроблено розкриття хоча б одного з пунктів, вказаних вище (якщо він явно потрібний для вичерпного розкриття питання) або, якщо:

– при розкритті змісту питання в цілому правильно за зазначеними вимогами зроблені окремі помилки під час: використання формул.

3. Відповідь на питання оцінюється **«задовільно»**, якщо:

– відносно відповіді на найвищий бал не зроблено розкриття чотирьох чи більше пунктів, зазначених у вимогах до нього (якщо вони явно потрібні для вичерпного розкриття питання);

– одночасно присутні чотири чи більше типів недоліків, які окремо характеризують критерій оцінки питання;

– висновки, зроблені під час відповіді, не відповідають правильним чи загально визначеним при відсутності доказів супротивного аргументами, зазначеними у відповіді;

– характер відповіді дає підставу стверджувати, що особа, яка складає іспит, не зовсім правильно зрозуміла зміст питання чи не знає правильної відповіді і тому не відповіла на нього по суті, допустивши грубі помилки у змісті відповіді.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
66-73	D	задовільно	
60-65	E		
30-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
1-29	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Загальна підсумкова оцінка в балах, за національною шкалою та за шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

11. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

- робоча навчальна програма дисципліни;
- електронний курс на платформі дистанційного навчання;
- плани лекцій, практичних занять та самостійної роботи студентів;
- тези лекцій з дисципліни;
- методичні вказівки до лабораторно-практичних занять для студентів;
- інструкції лабораторно-практичних занять
- методичні матеріали, що забезпечують самостійну роботу студентів;
- відеоматеріали платформи курсу «Створення та розвиток ІТ-продуктів»;

– перелік питань до заліку.

12. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна:

1. Лавріщева К.М. Програмна інженерія. Київ, 2008. 319 с.
2. Бородкіна І.Л., Бородкін Г.О. Інженерія програмного забезпечення. Навч. посібник. НУБіП. Київ: Центр учбов. літератури. 2020. 204 с.
3. Pfleeger S.L. Software Engineering. Theory and practice. Printice Hall: Upper Saddle River, New Jersey, 1998. 576 p.
4. Иан Соммервил. Инженерия программного обеспечения. 6-е издание. Пер. с англ. М.; Спб. Киев, 2002. 623 с.
5. Брукс Ф.П. Мифический человеко-месяц или как создаются программные системы. Пер.с англ. СПб.: Символ–Плюс, 2005. 304 с.
6. Основы инженерии качества программных систем / Ф.И.Андон, Г.И.Коваль, Т.М. Коротун, Е.М.Лаврищева, В.Ю. Суслов. Київ: Академперіодика. 2007. 678 с.

Додаткова:

1. Jacobson I. Object-Oriented Software Engineering. A use Case Driven Approach, Revised Printing. New York: Addison-Wesley Publ. Co., 1994. 529 p.
2. David C. Evans “Bottlenecks: Aligning UX Design with User Psychology”
3. Jon Yablonski “Laws of UX: Using Psychology to Design Better Products & Services”
4. Nir Eyal “Hooked. How to Build Habit-Forming Products”

Інформаційні ресурси

1. <https://www.uxmatters.com/mt/archives/2011/03/how-anchoring-ordering-framing-and-loss-aversion-affect-decision-making.php>
2. <https://www.uxmatters.com/mt/archives/2011/03/how-anchoring-ordering-framing-and-loss-aversion-affect-decision-making.php>