

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра програмної інженерії та інформаційних технологій управління
(назва)

«ЗАТВЕРДЖУЮ»

Голова науково-методичної комісії _____
(назва комісії)

_____ (підпис) _____ (ініціали та прізвище)

« _____ » _____ 20 _____ року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ (ЧАСТИНА 2)
_____ (назва навчальної дисципліни)

рівень вищої освіти перший (бакалаврський)
перший (бакалаврський) / другий (магістерський)

галузь знань 12 Інформаційні технології
(шифр і назва)

спеціальність 126 Інформаційні системи та технології
(шифр і назва)

спеціалізація _____
(шифр і назва)

вид дисципліни професійна підготовка
(загальна підготовка / професійна підготовка)

форма навчання денна
(денна / заочна)

Харків – 20 17 рік

ЛИСТ ЗАТВЕРДЖЕННЯ

Робоча програма з навчальної дисципліни
ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ (ЧАСТИНА 2)
(назва дисципліни)

Розробники:

<u>старший викладач</u>	_____	<u>Лютенко І.В.</u>
(посада, науковий ступінь та вчене звання)	(підпис)	(ініціали та прізвище)
_____	_____	_____
(посада, науковий ступінь та вчене звання)	(підпис)	(ініціали та прізвище)

Робоча програма розглянута та затверджена на засіданні кафедри
програмної інженерії та інформаційних технологій управління
(назва кафедри)

Протокол від « 31 » серпня 2017 року № 1

Завідувач кафедри ПШТУ _____ проф. М.Д. Годлевський
(назва кафедри) (підпис) (ініціали та прізвище)

ЛИСТ ПЕРЕЗАТВЕРДЖЕННЯ РОБОЧОЇ НАВЧАЛЬНОЇ ПРОГРАМИ

Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри	Підпис голови НМК (для дисциплін загальної підготовки та дисциплін професійної підготовки за спеціальністю) або завідувача випускової кафедри (для дисциплін професійної підготовки зі спеціалізації, якщо РПНД розроблена не випусковою кафедрою)

МЕТА, КОМПЕТЕНТНОСТІ, РЕЗУЛЬТАТИ НАВЧАННЯ ТА СТРУКТУРНО-ЛОГІЧНА СХЕМА ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета навчальної дисципліни «Об'єктно-орієнтоване програмування (частина 2)» – це засвоєння необхідних знань з опанування сучасними технологіями аналізу, проектування та об'єктно-орієнтованої розробки програмних систем, вивчення складових частин об'єктно-орієнтованої парадигми, об'єктно-орієнтованої моделі та синтаксису мови об'єктно-орієнтованого програмування Java, а також застосування об'єктно-орієнтованого підходу, а також методів узагальненого програмування до проектування і реалізації програмного забезпечення.

Компетентності:

Загальні компетентності:

- Здатність до абстрактного мислення, аналізу та синтезу.
- Здатність застосовувати знання у практичних ситуаціях.
- Здатність приймати обґрунтовані рішення.
- Здатність вчитися і оволодівати сучасними знаннями.
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові) компетентності:

- Здатність приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
- Здатність розробляти архітектури, модулі та компоненти програмних систем.

Нормативний зміст підготовки здобувачів вищої освіти, сформульований у термінах результатів навчання

Класифікація компетентностей за НРК	Знання	Уміння	Комунікація	Автономія та відповідальність
Здатність приймати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. Здатність розробляти архітектури, модулі та компоненти програмних систем	Знання етапів розвитку об'єктно-орієнтованого підходу, його складових частин та принципів побудови; вимог сучасних стандартів та мов об'єктно-орієнтованого моделювання, синтаксис та об'єктно-орієнтовану модель мови Java	Вміння використовувати сучасні об'єктно-орієнтовані технології для проектування та реалізації програмного забезпечення; володіти навичками програмування у сучасних середовищах програмування	Здатність ефективно формувати комунікаційні стратегії в сфері об'єктно-орієнтованого аналізу, проектування і програмування	Здатність самостійно здійснювати об'єктно-орієнтований аналіз і проектування систем, реалізовувати класи програмного забезпечення

Структурно-логічна схема вивчення навчальної дисципліни

Попередні дисципліни:	Наступні дисципліни:
Об'єктно-орієнтоване програмування (частина 1)	Основи проектування інформаційних систем
Алгоритмізація та програмування	Основи веб-технологій
Алгоритми та структури даних	Програмування веб-застосунків
Архітектура обчислювальних систем	Сучасні технології автоматизованої обробки інформації

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

(розподіл навчального часу за семестрами та видами навчальних занять)

Семестр	Загальний обсяг (годин) / кредитів ECTS	з них		За видами аудиторних занять (годин)			Індивідуальні завдання студентів (КП, КР, РГ, Р, РЕ)	Поточний контроль	Семестровий контроль	
		Аудиторні заняття (годин)	Самостійна робота (годин)	Лекції	Лабораторні заняття	Практичні заняття, семінари			Контрольні роботи (кількість робіт)	Залік
1	2	3	4	5	6	7	8	9	10	11
4	120/4	80	40	32	48			1		+

Співвідношення кількості годин аудиторних занять до загального обсягу складає 66% (%):

СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ з/п.	Види навчальних занять (Л, ЛЗ, ПЗ, СР)	Кількість годин	Номер семестру (якщо дисципліна викладається у декількох семестрах). Назви змістових модулів. Найменування тем та питань кожного заняття. Завдання на самостійну роботу.	Рекомендована література (базова, допоміжна)
1	2	3	4	5
Змістовий модуль 1. Синтаксис мови об'єктно-орієнтованого програмування Java				
	Л	8	Тема 1. Використання успадкування та поліморфізму в Java Концепція та синтаксис успадкування й поліморфізму. Робота з функціональними інтерфейсами Тема 2. Робота з узагальненнями в Java Концепція і синтаксис узагальнень. Використання узагальнень для створення універсальних контейнерів Тема 3. Використання рефлексії та мета програмування Використання RTTI та рефлексії. Завантажувачі класів	
	ЛР	12		
	СР	8		
	Л	6		
	ЛР	12		
	СР	8		
	Л	6		
	ЛР	10		
	СР	8		
Змістовий модуль 2. Основи об'єктно-орієнтованого проектування				
	Л	8	Тема 4. Об'єктно-орієнтоване моделювання і проектування Основні засади об'єктно-орієнтованого моделювання і проектування. Використання уніфікованої мови моделювання (UML) Тема 5. Використання патернів проектування Основи опису та використання патернів проектування	
	ЛР	10		
	СР	8		
	Л	4		
	ЛР	4		
	СР	8		
Разом (годин)		120		

САМОСТІЙНА РОБОТА

№ з/п	Назва видів самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу	16
2	Підготовка до практичних(лабораторних, семінарських) занять	16
3	Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях	4
4	Виконання курсової роботи	
5	Інші види самостійної роботи	4
	Разом	40

ІНДИВІДУАЛЬНІ ЗАВДАННЯ

(вид індивідуального завдання)

МЕТОДИ НАВЧАННЯ

При викладанні навчальної дисципліни для активізації навчального процесу передбачено застосування сучасних навчальних технологій, таких, як: проблемні лекції; робота в малих групах; семінари-дискусії; кейс-метод; ділові ігри.

Проблемні лекції спрямовані на розвиток логічного мислення студентів. Коло питань теми лекції обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов широкого відображення в підручниках, використовується досвід закордонних навчальних закладів. При викладанні лекційного матеріалу студентам пропонуються питання для самостійного розмірковування. При цьому лектор ставить запитання, які спонукають студента шукати розв'язання проблемної ситуації. Така система примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді.

На початку проведення проблемної лекції необхідно чітко сформулювати проблему, яку необхідно вирішити студентам. При викладанні лекційного матеріалу слід уникати прямої відповіді на поставлені запитання, а висвітлювати лекційний матеріал таким чином, щоб отриману інформацію студент міг використовувати при розв'язанні проблеми.

Міні-лекції передбачають викладання навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження. На початку проведення міні-лекції за вказаними темами лектор акцентує увагу студентів на необхідності представити викладений лекційний матеріал у так званому структурно-логічному вигляді. На розгляд виносяться питання, які зафіксовані у плані лекцій, але викладаються вони стисло. Лекційне заняття, проведене у такий спосіб, пробуджує у студента активність та увагу при сприйнятті матеріалу, а також спрямовує його на використання системного підходу при відтворенні інформації, яку він одержав від викладача. Проблемні лекції та міні-лекції доцільно поєднувати з такою формою активізації навчального процесу, як робота в малих групах.

Робота в малих групах дає змогу структурувати лекційні або лабораторні заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування. Після висвітлення проблеми (при використанні проблемних лекцій) або стислого викладання матеріалу (при використанні міні-лекцій) студентам пропонується об'єднуватися у групи по 5-6 осіб та презентувати наприкінці заняття своє бачення та сприйняття матеріалу.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань. Однією з позитивних рис презентації та її переваг при використанні в навчальному процесі є обмін досвідом, який здобули студенти при роботі у певній малій групі.

Лабораторні заняття (з елементами семінарської дискусії) дозволяють формувати у студентів навички особистого експериментального дослідження фізичних процесів що відбуваються під час роботи компонентів операційної системи, проводити аналіз умов її функціонування, а також розробляти нові елементи та системні компоненти відповідно до вимог, що пред'являються до них, узагальнювати отримані результати, формулювати висновки та думки, вести подальший обмін думками та поглядами з іншими учасниками щодо отриманих результатів досліджень з даної теми, а також розвивають творче мислення, допомагають формувати погляди і переконання, вчать об'єктивно оцінювати результати і пропозиції опонентів, критично підходити до власних результатів та поглядів.

Ділові та рольові ігри – форма активізації студентів, за якої вони задіяні в процесі інсценізації певної виробничої ситуації у ролі безпосередніх учасників подій. Наприклад, при проведенні лабораторного заняття за темою "Дослідження організації пам'яті в захищеному режимі" слід поділити аудиторію на групи, кожній з яких дати завдання використовуючи різноманітні алгоритми розподілу пам'яті визначити ступінь ефективності використання фізичної пам'яті комп'ютера.

Кейс-метод – метод аналізу конкретних ситуацій, який дає змогу наблизити процес навчання до реальної практичної діяльності спеціалістів і передбачає розгляд виробничих, управлінських та інших ситуацій, складних конфліктних випадків, проблемних ситуацій, інцидентів у процесі вивчення навчального матеріалу.

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни наведено у табл. 4.

Тема	Практичне застосування навчальних технологій
Тема 1. Використання успадкування та поліморфізму в Java	Проблемна лекція, з питання "Порівняння об'єктно-орієнтованих моделей Java і C++";
Тема 2. Робота з узагальненнями в Java	Міні-лекція, лабораторне заняття (семінар-дискусія) з питання "Загальні функції контейнерів"; презентація результатів роботи в малих групах

Тема 3. Використання рефлексії та мета програмування	Міні-лекція, лабораторне заняття (семінар-дискусія) з питань "Інтерпретація й компіляція коду в мета програмуванні";
Тема 4. Об'єктно-орієнтоване моделювання і проектування	Проблемна лекція з питання "Ієрархія класів та ієрархія об'єктів"; презентація результатів роботи в малих групах
Тема 5. Використання патернів проектування	Міні-лекція, лабораторне заняття (семінар-дискусія) з питання "Використання структурних патернів"

МЕТОДИ КОНТРОЛЮ

Система оцінювання знань, вмінь та навичок студентів передбачає виставлення оцінок за усіма формами проведення занять. Перевірка та оцінювання знань студентів може проводитись у таких формах:

1. Оцінювання роботи студентів у процесі лабораторних занять.
2. Проведення проміжного контролю.
3. Проведення модульного контролю.

Загальна модульна оцінка складається з поточної оцінки, яку студент отримує під час лабораторних занять та оцінки за виконання модульної контрольної роботи.

Загальна оцінка з дисципліни визначається як середнє арифметичне модульних оцінок.

Порядок поточного оцінювання знань студентів

Поточне оцінювання здійснюється під час проведення лабораторних занять і має на меті перевірку рівня підготовленості студента до виконання конкретної роботи. Об'єктами поточного контролю є:

- 1) активність та результативність роботи студента протягом семестру над вивченням програмного матеріалу дисципліни; відвідування занять;
- 2) виконання проміжного контролю;
- 3) виконання модульного контрольного завдання.

Контроль систематичного виконання самостійної роботи та активності на лабораторних заняттях

Оцінювання проводиться за 5-бальною шкалою за такими критеріями:

- 1) розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються;
- 2) ступінь засвоєння матеріалу дисципліни;

- 3) ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;
- 4) уміння поєднувати теорію з практикою при розгляді виробничих ситуацій, розв'язанні задач, проведенні розрахунків при виконанні завдань, винесених для самостійного опрацювання, та завдань, винесених на розгляд в аудиторії;
- 5) логіка, структура, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" ставиться за умови відповідності виконаного завдання студента або його усної відповіді до всіх п'яти зазначених критеріїв.

Відсутність тієї чи іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні практичних завдань увага приділяється також їх якості та самостійності, своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу). Якщо якась із вимог не буде виконана, то оцінка буде знижена.

Проміжний модульний контроль

Проміжний модульний контроль рівня знань передбачає виявлення опанування студентом матеріалу лекційного модуля та вміння застосовувати його для вирішення практичної ситуації і проводиться у вигляді тестування. При цьому тестове завдання може містити як запитання, що стосуються суто теоретичного матеріалу, так і запитання, спрямовані на вирішення невеличкого практичного завдання.

Тестове завдання містить запитання одиничного і множинного вибору різного рівня складності. Для оцінювання рівня відповідей студентів на тестові завдання використовуються такі критерії оцінювання:

- оцінка "відмінно" (12 – 10 балів) – виставляється у випадку, якщо студент правильно відповів на 24 – 20 тестових запитань;
- оцінка "дуже добре" (9 балів) – 19 – 18 правильних відповідей;
- оцінка "добре" (8 – 7 балів) – 17 – 14 правильних відповідей;
- оцінка "задовільно" (6 балів) – 13 – 12 правильних відповідей;
- оцінка "достатньо" (5 – 4 балів) – 11 – 8 правильних відповідей;
- оцінка "незадовільно" (3 бали) – 7 – 6 правильних відповідей;
- оцінка "незадовільно" (2 – 1 бали) – 5 – 0 правильних відповідей.

Тести для проміжного контролю обираються із загального переліку тестів за відповідними модулями.

Проведення модульного контролю

Модульний контроль здійснюється та оцінюється за двома складовими: лекційний (теоретичний) модуль та практичний модуль.

Теоретичний модульний контроль проводиться у письмовій формі після того як розглянуто весь теоретичний матеріал. Після вивчення тем 1 – 3 (модуль 1) студенти денної форми навчання виконують – завдання до модуля 1. Відповідно, після вивчення тем 4 – 5 (модуль 2) – завдання до модуля 2.

Практичний модульний контроль проводить після виконаних лабораторних завдань у межах кожного з двох модулів з урахуванням захищених звітів з лабораторних робіт.

Теоретичне модульне завдання оцінюється за 12-бальною системою відповідно до кваліфікаційних вимог до бакалаврів напряму підготовки 6.050101 "Комп'ютерні науки". При цьому вважається, що для набору 1 бала оцінки потрібно правильно відповісти на 2 запитання завдання до модуля.

Загальна оцінка за практичне модульне завдання визначається як середнє арифметичне з оцінок за виконані лабораторні роботи при округленні в сторону студента.

Підсумкова оцінка з дисципліни розраховується як середня з кількох складових, що враховує оцінки кожного виду контролю (дві оцінки за результатами поточного модульного контролю за роботу протягом семестру).

РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ, ТА ШКАЛА ОЦІНЮВАННЯ ЗНАНЬ ТА УМІНЬ (НАЦІОНАЛЬНА ТА ECTS)

Таблиця 1 – Розподіл балів

	Поточний контроль			Семестровий контроль	Всього за семестр
	КР	ЛР	Опитування		
Підсумкові бали	85			15	100
Макс. проміжні бали	25	10	10		
Кільк. од. обліку у семестрі	1	5	1		
Макс. проміжних балів, всього	25	50	10		100
Коеф.. перерахунку	1				
Макс. кільк. підсумкових балів	25	25	35	15	100

Таблиця 2. Шкала оцінювання знань та умінь: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 ... 100	A	відмінно
82 ... 89	B	добре
74 ... 81	C	
64 ... 73	D	задовільно
60 ... 63	E	
35 ... 59	FX	незадовільно з можливістю повторного складання
0 ... 34	F	незадовільно з обов'язковим повторним вивченням дисципліни

НАВЧАЛЬНО-МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Іванов Л.В. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування (перша частина)» [Електронний ресурс]. – Режим доступу: iwanoff.inf.ua

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базова

1. Шилдт, Г. Java 8. Полное руководство; 9-е изд. – М. : ООО "И.Д.Вильямс", 2015. – 1376 с. : ил.
2. Хорстманн К. С., Корнелл Г. Java 2. Библиотека профессионала, том 1. Основы, 7-е изд. – М. : Издательский дом "Вильямс", 2007. – 896 стр. с.
3. Шилдт Г., Холмс Д. Искусство программирования на Java. – М.: Издательский дом "Вильямс", 2005. – 336 с.
4. Эккель Б. Философия Java. Библиотека программиста. 4-е изд. – СПб.: Питер, 2009. – 640 с.
5. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с.
6. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с.
7. Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: Навч. посібник. - Львів: Видавничий Центр ЛНУ імені Івана Франка, 2007. - 108 с.

8. Буч Г., Рамбо Дж. Джекобсон А. Язык UML. Руководство пользователя: Пер. с англ. – М.: ДМК, 2000. - 432 с.
9. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2001.

Допоміжна література

1. Хорстманн К. С., Корнелл Г. Java 2. Библиотека профессионала, том 2. Тонкости программирования, 7-е изд. – М.: Издательский дом "Вильямс", 2007. – 1168 стр. с.
2. Дейтел Х.М., Дейтел Как программировать на Java. Книга 2. Файлы, сети, базы данных. – М.: Бином, 2006. – 672 с.
3. Блинов, И.Н., Романчик В.С. Java. Промышленное программирование : практ. пособие. – Минск : УниверсалПресс, 2007. – 704 с.
4. Фаулер М., Скотт К. UML. Основы / Пер. с англ. – СПб: Символ-Плюс, 2002.

ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. Освоюємо Java – Вікіпідручник // http://uk.wikibooks.org/wiki/Освоюємо_Java
2. Програмування на Java // <http://javaland.com.ua>
3. Брнакевич І.С., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій // http://blues.franko.lviv.ua/ami/books/ami/Java_fundamental.pdf
4. Начинаящим Java программистам // <http://habrahabr.ru/blogs/java/43293/> (рос.)
5. Java Tutorials // <http://docs.oracle.com/javase/tutorial> (англ.)
6. Java Tutorial // <http://www.java2s.com/Tutorial/Java/CatalogJava.htm>
7. Bruce Eckel. Thinking in Java, 4th Edition // http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf (англ.)
8. IT – archiv :: Information technology community // <http://www.javable.com>
9. Java programming notes // <http://leepoint.net/notes-java>
10. Unified Modeling Language – Вікіпедія // http://uk.wikipedia.org/wiki/Unified_Modeling_Language
11. UML: історія, специфікація, бібліографія // <http://it.ridne.net/node/265>
12. UML Resource Page // <http://www.uml.org/>
13. Шаблиони проектування програмного забезпечення – Вікіпедія // https://uk.wikipedia.org/wiki/Шаблиони_проектування_програмного_забезпечення
14. Design Patterns: Elements of Reusable Object-Oriented Software // <http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf>