

Одеський національний університет імені І.І. Мечникова
Кафедра теоретичної механіки

Проректор

“ЗАТВЕРДЖУЮ”

“ _____ 20__ р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК 15 «Об’єктно-орієнтоване програмування»

(назва навчальної дисципліни)

Рівень вищої освіти _____ бакалавр _____

Спеціальність _____ 126 – Інформаційні системи та технології _____
(код і назва спеціальності (тей))

Факультет _____ математики, фізики та інформаційних технологій _____
(назва інституту, факультету)

Робоча програма складена на основі навчальної програми з дисципліни «Об'єктно-орієнтоване програмування».

Розробник:

Рачинська А.Л., к.ф.-м.н., доцент кафедри ТМ

Робоча програма затверджена на засіданні кафедри теоретичної механіки

Протокол № 1 від. "31" серпня 2021 р.

Завідувач кафедри

(підпис)

(В.Е. Волков)
(прізвище та ініціали)

Схвалено навчально-методичною комісією (НМК) з ІТ факультету МФІТ

Протокол № 1 від. "6" 09 2021 р.

Голова НМК

(підпис)

(О.С. Савастру)
(прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ____ від. " " _____ 20__ р.

Завідувач кафедри

(підпис)

(_____)
(прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ____ від. " " _____ 20__ р.

Завідувач кафедри

(підпис)

(_____)
(прізвище та ініціали)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Загальна кількість: кредитів 8 ECTS годин 240 залікових модулів 1 змістових модулів 3 ІНДЗ* – _____ (вид завдання)	Галузь знань <i>12 Інформаційні технології</i> (шифр і назва) Спеціальність <i>126 Інформаційні системи та технології</i> (код і назва) Спеціалізації: _____ (назва) Рівень вищої освіти: <i>бакалавр</i>	Нормативна / за вибором (ВНЗ/студента)	
		Рік підготовки:	
		1-й	1-й
		Семестр	
		1-й	2-й
		Лекції	
		64 год.	22 год.
		Практичні, семінарські	
		год.	год.
		Лабораторні	
		64 год.	22 год.
		Самостійна робота	
		112 год.	196 год.
		у т.ч. ІНДЗ*: - год.	
		Форма підсумкового контролю: <i>іспит</i>	

* – за наявності

2. Мета та завдання навчальної дисципліни

Мета

Мета викладання дисципліни *«Об'єктно-орієнтоване програмування»* полягає у наданні теоретичних знань та практичних навиків програмування з використанням об'єктно-орієнтованих мов програмування. Мета проведення лекцій полягає у тому, щоб ознайомити студентів із головними методологічними та методичними питаннями об'єктно-орієнтованої парадигми програмування, та інструментарієм для створення об'єктно-орієнтованих програм від моделювання взаємозв'язків між об'єктами до створення програмного коду засобами об'єктно-орієнтованої мови програмування C#. Мета проведення лабораторних занять полягає у тому, щоб виробити у студентів практичні навички побудови об'єктно-орієнтованих моделей та їх програмної реалізації засобами об'єктно-орієнтованої мови програмування C#.

Завдання:

Завданням дисципліни є набуття студентами теоретичних знань та практичних навичок принципів об'єктно-орієнтованої декомпозиції складних об'єктів, розробки проектів та написанні на їх основі ефективних програм з використанням всіх переваг об'єктно-орієнтованої парадигми програмування.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей**:

а) Загальні компетентності КЗМЗ, КЗМ4.

КЗМ 3. Здатність розуміти і продукувати ділову кореспонденцію (напр., листи, факси, електронні повідомлення тощо).

КЗМ 4. Здатність готувати повідомлення, доповіді, презентації відповідно до загальноприйнятої структури та використовуючи мовні форми, властиві для усного професійного мовлення

Спеціальні (фахові, предметні) компетентності: КС1, КС4, КС12, КСМ5.

КС 1. Здатність аналізувати об'єкт проектування або функціонування та його предметну область.

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

КС 12. Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).

КСМ 5. Здатність виконувати інформаційне моделювання предметних областей інформаційних систем, проектувати логічні та фізичні моделі баз даних та створювати запити до них.

Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен отримати наступні ПРН:

ПР 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм

мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

ПР 4. Проводити системний аналіз об'єктів проектування та обґрунтовувати вибір структури, алгоритмів та способів передачі інформації в інформаційних системах та технологіях.

ПР 5. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем та технологій.

ПР 6. Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ з метою їх запровадження у професійній діяльності.

ПРМ 3. Застосовувати методи та алгоритми комп'ютерної графіки у процесі розробки графічних застосувань, систем мультимедіа, також графічного моделювання та візуалізації фізичних процесів і об'єктів.

У результаті вивчення дисципліни студенти повинні вміти: – визначати класи та об'єкти при застосуванні об'єктно-орієнтованих технологій програмування; – здійснювати об'єктно-орієнтовану декомпозицію предметної області в умовах програмування складних об'єктів і систем за допомогою процедур об'єктно-орієнтованого аналізу, об'єктно-орієнтованого програмування, використовуючи визначення класів, поведінки об'єктів, структури даних та їх взаємозв'язки; – розробляти об'єктно-орієнтовану модель предметної галузі за допомогою мов об'єктно-орієнтованого моделювання в умовах програмування складних об'єктів та систем, використовуючи об'єктно-орієнтовану нотацію складних систем.

знати:

основні принципи розробки об'єктно-орієнтованих програм з використанням мови програмування C#;

основні принципи об'єктного підходу та взаємозв'язків об'єктно-орієнтованого аналізу, об'єктно-орієнтованого проектування і об'єктно-орієнтованого програмування;

вміти:

використовувати основні засоби об'єктно-орієнтованої моделі - інкапсуляцію, спадкування поліморфізм при створенні програм з використанням мови програмування C#;

використовувати навички обробки об'єктно-орієнтованих даних.

3. Зміст навчальної дисципліни

Змістовий модуль 1. Принципи ООП

Тема 1. Налагодження програми і обробка помилок.

Тема 2. Перший принцип ООП. Визначення класів в C #.

Тема 3. Об'єкти і класи.

Тема 4. Часткові і статичні класи.

Тема 5. Перевантаження операторів.

Тема 6. Другий принцип ООП. Спадкування.

Тема 7. Третій принцип ООП. Підтримка поліморфізму в C#.

Тема 8. Інтерфейси.

Тема 9. Обробка виняткових ситуацій.

Змістовий модуль 2. Технологія WFA.

Тема 1. Технологія Windows Forms Application.

Тема 2. Форма та стандартні компоненти.

Тема 3. Контейнери.

Тема 4. Події клавіатури та миші.

Тема 5. Вікно повідомлень.

Тема 6. Створення і використання динамічних компонент.

Тема 7. Використання діалогових вікон.

Тема 8. Керуючі елементи ListBox, ComboBox та PictureBox.

Тема 9. Створення бібліотек.

Тема 10. Побудова графіків.

Тема 11. Технологія Drag & Drop.

Тема 12. Делегати і події.

Тема 13. Створення керуючих елементів.

Тема 14. Елемент управління DataGridView.

Змістовий модуль 3. Узагальнення знань.

Тема 1. Серіалізація

Тема 2. UML – діаграми.

4. Структура навчальної дисципліни

Назви тем	Кількість годин			
	Денна форма		Заочна форма	
	Усього	у тому числі	Усього	у тому числі

		л	п/с	лаб	ср		л	п/с	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовий модуль 1. Принципи ООП										
Тема 1. Налагодження програми і обробка помилок	10	2		2	6	10	1			9
Тема 2. Перший принцип ООП. Визначення класів в C #	5	2			3	5	1			4
Тема 3. Об'єкти і класи	9	2		2	5	9	2		1	6
Тема 4. Часткові і статичні класи	5	2			3	5	1			4
Тема 5. перевантаження операторів	10	4		4	2	10	1		2	7
Тема 6. Другий принцип ООП. Спадкування	5	2			3	5	1			4
Тема 7. Третій принцип ООП. Підтримка поліморфізму в C#	10	4		2	4	10	2		1	7
Тема 8. Інтерфейси.	5	4			1	5				5
Тема 9. Обробка виняткових ситуацій.	5	4			1	5				5
Разом за змістовим модулем 1	64	26		10	28	64	9		4	31
Змістовий модуль 2. Технологія WFA.										
Тема 1. Технологія Windows Forms	12	3		4	5	12	1			11

Application.										
Тема 2. Форма та стандартні компоненти.	12	2		4	6	12	1		2	9
Тема 3. Контейнери.	10	2		4	4	10	1		2	7
Тема 4. Події клавіатури та миші.	10	2		4	4	10	1		2	7
Тема 5. Вікно повідомлень	10	2		4	4	10	1		2	7
Тема 6. Створення і використання динамічних компонент	14	3		6	5	14	1		2	11
Тема 7. Використання діалогових вікон.	12	2		4	6	12	1		2	9
Тема 8. Керуючі елементи ListBox, ComboBox та PictureBox	10	2		4	4	10	1		2	7
Тема 9. Створення бібліотек.	10	2		4	4	10	1		2	7
Тема 10. Побудова графіків.	12	2		4	6	12				12
Тема 11. Технологія Drag & Drop.	14	2		4	8	14	1		2	11
Тема 12. Делегати і події	10	2		4	4	10	1			9
Тема 13. Створення керуючих елементів	12	3		4	5	12	1			11
Тема 14. Елемент управління DataGridView	12	3			9	12				12
Разом за змістовим модулем 2	160	32		54	74	160	12		18	50
Змістовий модуль 3. Узагальнення знань										
Тема 1. Сериалізація	10	4			6	10				10
Тема 2. UML –	6	2			4	6	1			5

діаграми										
Разом за змістовим модулем 3	16	6		0	10	16	1			15
ІНДЗ*										
Усього годин	240	64		64	112	240	22		22	196

* – Не передбачено робочим навчальним планом

5. Теми семінарських занять

Не передбачено навчальним планом

6. Теми практичних занять

Не передбачено навчальним планом

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Масиви масивів	4
2	Списки	4
3	Створення власного класу	4
4	Перевантаження операторів	4
5	Масиви і класи	4
6	Спадкування та поліморфізм	4
7	Вивчення принципів візуального програмування	8
8	Робота з текстовою інформацією	8
9	Динамічні компоненти	8
10	Обробка інформації, що зберігається в файлах	8
11	Технологія Drag & Drop	8
12	Створення власного пакета компонента	8

8. Самостійна робота

№ з/п	Назва теми/ види завдань	Кількість годин
1	Гра «Морський бій» / індз	50
2	Гра «Тренування пам'яті» / індз	50
3	Гра «Нім» / індз	50
4	Гра в слова/ індз	50
5	Гра «Вгадай число» / індз	50
6	Гра «Кості» / індз	50
7	Гра «Корови і бики» / індз	50
8	Гра «Від одного до десяти» / індз	50
9	Гра «Хрестики - нулики» / індз	50

10	Гра «Фруктова машина» / індз	50
11	Гра «Судоку» / індз	50
12	Гра «Сапер» / індз	50
13	Гра «Японський кросворд» / індз	50
14	Гра «Парні картинки» / індз	50
15	Гра «Доміно-телефон» / індз	50

1. Гра «Морський бій»

Запрограмувати відому гру «Морський бій». На формі поле невеликого розміру – поле гравця. Поле комп'ютера в невидиме (після завершення гри становиться видимим). На полі 10×10 стоять невидимі ворожі пароплави: 4 корабля по 1 клітинці, 3 корабля по 2 клітинки, 2 корабля по 3 клітинки, 1 корабель в 4 клітинки.

Необхідно поцілити кожну з клітин корабля. Позиції вказують буквами від А до К (по рядах) і цифрами від 1 до 10 (по стовпцям). Якщо клітинка вгадана гравцем вірно, вона помічається хрестиком, інакше точкою. Конфігурація і положення корабля на полі комп'ютера виставляються за допомогою датчика випадкових чисел. Гравець і комп'ютер по черзі роблять ходи. Повинно видатись повідомлення «попав», «вбитий», «виграли», «спочатку».

2. Гра «Тренування пам'яті»

Скласти програму для тренування пам'яті. Програма повинна висвітити на екрані декілька точок, а той, хто грає повинен вказати, в якому порядку ці точки були висвітлені. Координати точок вибираються в програмі з допомогою датчика випадкових чисел.

Для реалізації можна використовувати масив меток, в якому деякі метки замальовуються іншим кольором. Можна організувати різні рівні складності, змінюючи кількість компонент на формі та кількість точок висвітлення.

3. Гра «Нім»

Маємо три купки сірників. Двоє грають по черзі роблячи ходи. Кожний хід полягає в тому, що з однієї якоїсь купки береться довільна не рівна нулю кількість сірників. Вигріє той, хто візьме останній сірник.

Для реалізації в якості сірників можна використовувати масиви компонентів `pictureBox` з завантаженими зображеннями сірників. Компоненти повинні розташовуватися один на одному, щоб не можливо було вгадати їх кількість.

4. Гра в слова

Реалізувати гру «Відгадай слово», яка відбувається таким чином. Машина загадує слово(випадково вибирає його з наявного списку) і виводить на екран першу і останню букви і риски замість решти букв. Граючий на кожному ході намагається відгадати одну букву слова. Ведеться протокол гри і підраховується «коефіцієнт інтелекту»

$$КИ = \frac{a - b}{x}$$

де a - сумарне число названих букв,

b - число вгаданих букв,

x - первинне число невідомих букв.

Гра закінчується, коли вгадано слово або коли «коефіцієнт інтелекту» стає більше 2.

5. Гра «Вгадай число»

Один з гравців задумує число від 1 до 1000, другий пробує вгадати його за десять питань виду: чи вірно, що задумане число більше такого-то числа. Написати програму, яка грає за того, хто відгадує.

Розробити програму, яка буде виконувати це завдання для будь-якого задуманого числа. Передбачити обмеження числа ходів, а також штрафні санкції та призові виплати в вигляді премії або картинки призу.

6. Гра «Кості»

Гравець називає будь-яке число в діапазоні від 2 до 12 і ставку, яку він робить в цей хід. Програма за допомогою датчика випадкових чисел двічі вибирає число від 1 до 6 («кидає кубик», на гранях якого цифри від 1 до 6). Якщо сума цифр, що випали, менше 7 і гравець задумав число менше 7, він виграє зроблену ставку. Якщо сума цифр, що випали, більше 7 і гравець задумав число більше 7, він також виграє зроблену ставку. Якщо гравець вгадав суму цифр, він отримує в чотири рази більше очок, чим зроблена ставка. Ставка програна, якщо не має місце жодна із описаних ситуацій. В початковий момент у гравця 100 очок.

7. Гра «Корови і бики»

Програма вибирає за допомогою датчика випадкових чисел чотиризначне число з різними цифрами. Вгадати це число. На кожному кроці той, хто грає називає 4-х значне число, а програма повідомляє скільки цифр числа вгадано(бики) і скільки цифр вгадано і стоїть на потрібному місці (корови). Наприклад, якщо програмою задано число 1294, а той, хто грає назвав 1423, він отримав відповідь «1 корова, 3 бика». Непогано було б замість слів зобразити малюнки.

8. Гра «Від одного до десяти»

Пропонується створити простеньку ігрову програму для навчання дитини.

Ця нескладна гра допомагає малюку запам'ятати в якому порядку йдуть числа від 1 до 10. Питання, яке в кожному ході пропонує програма, виглядає так: «Яке число йде за n » (N – будь-яке число від 1 до 9). Навіть поки дитина ще не

вміє читати, зігравши з вашою допомогою в цю гру декілька разів, вона легко навчиться розуміти, що у нього питають.

Вірна відповідь нагороджується призом – розфарбованою картинкою. Невірна – викликає повторне завдання того ж самого питання. «Пуста відповідь» (клавіша «Введення» натиснута без будь-якого числового значення) приводить до завершення гри.

Для того, щоб гра стала більш привабливою, зробіть більш цікавим приз, котрим нагороджується правильна відповідь. Нехай можливих призів буде декілька. А вибрати, яким призом нагородити чергову правильну відповідь, - ваш датчик випадкових чисел.

9. Гра «Хрестики - нулики»

Написати програму, що реалізує гру «Хрестики-нулики». Програма передбачає гру між двома гравцями, які по черзі здійснюють хід по ігровому полі розміром 3×3 . Виграє той гравець, який першим одержав лінію з 3 хрестиків чи нуликів по горизонталі, вертикалі чи по діагоналі. Реалізувати можливість гри з комп'ютером. Гравці, в разі повного заповнення ігрового поля, мають змогу переграти або розпочати нову гру. Можна зробити різнорівневу гру.

10. Гра «Фруктова машина»

В цій «смачній» грі задіяні яблука, вишні, груші, сливи, малина і шоколадки. Грають так: ви робите ставку і за кожний хід (ходи робляться комп'ютером) або втрачаєте її, або до вас повертається більше, чим було поставлено. Успіх або невдача визначається тим, які три предмети з перерахованих вище були вибрані комп'ютером. Діють такі правила

- Три шоколадки дають гравцю весь банк,
- Будь-які три однакових фрукта повертають гравцю в 10 раз більше, чим зроблена ним ставка,
- Будь-які два однакових фрукта і одна шоколадка повертають гравцю в 5 разів більше, чим зроблена ним ставка,

- Дві шоколадки і будь-який фрукт повертають гравцю в 3 рази більше, чим зроблена ним ставка,
- У всіх інших випадках ставка програється.

11. Гра «Судоку»

Ігрове поле складається з квадрата, розміром 9×9 , розділеного на менші квадрати (їх ще називають «регіони») із стороною 3×3 клітинки. Таким чином, все поле налічує 81 клітинку. У деяких з них вже на початку гри розташовані числа (від 1 до 9). Залежно від того, скільки клітинок вже заповнено, конкретну «Судоку» можна віднести до легких або складних.

Мета головоломки - необхідно заповнити вільні клітинки цифрами від 1 до 9 так, щоб в кожному рядку, в кожному стовпці і в кожному малому квадраті 3×3 , кожна цифра зустрічалася лише один раз. Правильна головоломка має лише одне рішення.

12. Гра «Сапер»

Плоске або об'ємне ігрове поле поділене на суміжні комірки. Деякі з яких «заміновані»; кількість «замінованих» осередків відомо. Метою гри є відкриття всіх комірок, які не містять міни.

Гравець відкриває комірки, намагаючись не відкрити комірки з міною. Відкривши клітинку з міною, він програє. Міни розставляються після першого ходу, тому програти на першому ж ході неможливо. Якщо під відкритою коміркою міни немає, то в ній з'являється число, що показує, скільки комірок, що є сусідами з щойно відкритої, «заміновано» (в будь-якому вигляді гри сусідство визначається по-своєму); використовуючи ці числа, гравець намагається розрахувати розташування мін, однак іноді навіть в середині і в кінці гри деякі комірки все ж доводиться відкривати навмання. Якщо під сусідніми комірками теж немає мін, то відкривається деяка «Не замінована» область до комірок, в яких є цифри. «Заміновані» комірки гравець може помітити, щоб випадково не відкрити їх. Відкривши всі «Не заміновані» комірки, гравець виграє.

13. Гра «Японський кросворд»

Японський кросворд - головоломка, в якій за допомогою цифр зашифровано деяке зображення. Метою головоломки є повне відновлення цього зображення.

Японські кросворди діляться на два види - чорно-білі і кольорові. У чорно-білих кросвордах зображення містить тільки два кольори - чорний (яким ми і малюємо) і білий (колір фону). У кольорових кросвордах зображення створюється декількома квітами на білому тлі.

Поле японського кросворду розкреслений горизонтальними і вертикальними лініями різної товщини. Найтовстіші лінії відокремлюють центральну частину (поле для картинки) від цифр. Більш тонкими лініями, поле ділиться на групи по 5 клітин (як по горизонталі, так і по вертикалі) - це зроблено виключно для зручності (зручніше вважати ширину / висоту груп клітин). Саме зображення в японському кросворді формується шляхом зафарбовування окремих клітин (центральної частини) в потрібний колір. Чи не зафарбована клітина при цьому вважається білою.

Цифри, вказані зліва і зверху кросворду - описують групи зафарбованих клітин (що йдуть підряд, без пропусків) по горизонталі і вертикалі відповідно. Причому порядок цих цифр описує порядок розташування цих груп, але де кожна група починається і закінчується - невідомо (фактично, визначити їх положення і є завданням головоломки). Кожна окрема цифра позначає окрему групу заданого розміру (тобто цифра 5 - позначає групу з п'яти зафарбованих поспіль клітин, 1 - групу з однієї єдиної зафарбованою клітини). У чорно-білих кросвордах ми завжди зафарбовує клітку чорним кольором, в кольорових - зафарбовує клітку тим кольором, яким позначена цифра. Між групами одного кольору повинна бути як мінімум одна не зафарбована клітина (просто інакше вони вважалися б однією групою), між групами різних кольорів порожніх клітин може і не бути.

Для реалізації можна використовувати масив меток, в якому метки замальовуються вибраним кольором.

14. Гра «Парні картинки»

Відкривається таблиця з парної кількості клітин. У кожній з них захована картинка. За один хід ви можете розкрити дві картинки. Якщо картинки виявляться різними, вони знову сховаються, а якщо в обох відкритих клітках буде

одна і та ж картинка, то ці клітини залишаються відкритими. Гра буде закінчена, коли ви розкриєте всі клітини. Завдання користувача - зробити це якомога швидше. Можна зробити різнорівневу гру.

Для реалізації можна використовувати масиви компонентів pictureBox.

15. Гра «Доміно-телефон»

https://www.gambler.ru/Domino_five

До самостійної роботи відноситься:

- [1] – підготовка до лекцій, лабораторних занять;
- [2] – написання коду додатків лабораторних робіт;
- [3] – проведення аналізу отриманих моделей.

9. Індивідуальне навчально-дослідне завдання

Не передбачено

10. Методи навчання

Лекції з використанням мультимедійного презентаційного матеріалу.

11. Методи контролю

Під час **підсумкового контролю** студент повинен зробити індивідуальне завдання, наведене у п. 12.

Критерії оцінювання на підсумковому модульному контролі:

1. Відповідь повинна бути повною і короткою. Вона не повинна мати в собі матеріал, що не відноситься до сутті питання.
2. Чітко формулювати твердження, вправно застосовувати необхідні формули і знання основних питань програми.
3. Відповіді, що мають помилкові твердження оцінюються виходячи з близькості відповіді до правильної.
4. Пропуски в обґрунтуванні тверджень враховуються і це призводить до зменшення кількості балів.
5. Малі недоліки, неточності при викладенні матеріалу, зменшують кількість балів.
6. Незнання і нерозуміння основної ідеї теоретичного питання або задачі призводить до зняття до 90 % балів.
7. Якщо відповідь на питання відсутня то виставляється нуль балів.

12. Завдання для підсумкового контролю

1. Створити додаток для вирішення рівнянь виду $ax^2 + bx + c = 0$.
2. Створити додаток для обліку успішності групи студентів.
3. Створити додаток «Футбол».
4. Створити додаток-гру, яка має функціональні можливості: за допомогою технології Drag & Drop організувати перенесення картинок в порожні клітинки.
5. Створити додаток «Перехоплювач».
6. Створити додаток-гру «Числа».
7. Створити додаток для побудови графіків функцій.
8. Створити додаток для ілюстрування таблиці множення.
9. Створити додаток для розподілення дисциплін викладачам.
10. Створити додаток для створення динамічних компонентів.
11. Створити додаток-гру «Космічні кораблі».
12. Створити додаток для ілюстрування трикутника Паскаля.
13. Створити додаток для розподілення дисциплін студентам.
14. Створити додаток для вирішення рівнянь виду $ax^3 + bx^2 + cx = 0$.

13. Розподіл балів, які отримують студенти

Орієнтовний приклад для заліку

ЛР1	ЛР2	ЛР3	ЛР4	ЛР5	ЛР6	КР1	
5	5	5	5	5	5	10	
ЛР7	ЛР8	ЛР9	ЛР10	ЛР11	ЛР12	КР2	Всього
5	8	8	8	8	8	15	100

ЛР1-ЛР12 – лабораторні роботи

КР1, КР2 – контрольні роботи

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
85-89	B	добре	
75-84	C		
70-74	D	задовільно	
60-69	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов’язковим повторним	не зараховано з обов’язковим

		вивченням дисципліни	повторним вивченням дисципліни
--	--	----------------------	--------------------------------

13. Методичне забезпечення

1. Електронний конспект лекцій за дисципліною «Об'єктно - орієнтоване програмування» Classroom:
<https://drive.google.com/drive/folders/1LzkQSUk0nJchks5jJ-vlVwREjC5tiTwO?usp=sharing>
2. Лабораторні роботи за дисципліною «Об'єктно - орієнтоване програмування» Classroom:
<https://drive.google.com/drive/folders/1LzkQSUk0nJchks5jJ-vlVwREjC5tiTwO?usp=sharing>

14. Рекомендована література

Основна

1. Електронний конспект лекцій за дисципліною «Об'єктно - орієнтоване програмування» Classroom:
<https://drive.google.com/drive/folders/1LzkQSUk0nJchks5jJ-vlVwREjC5tiTwO?usp=sharing>

Додаткова

1. Бублик В.В. Об'єктно-орієнтоване програмування : ІТ-книга, 2015.
2. Шилдт Герберт. С# 4.0: полное руководство. : Пер. с англ. – М.: ООО «И. Д. Вильямс», 2011.
3. Мартынов Н. Н. С# для начинающих / Н. Н Мартынов . – М.: КУДИЦ-ПРЕСС, 2007.
4. Фомин Г.В. Введение в программирование на С# в среде MS Visual Studio. : ЮФУ, 2011.
5. Лабор В.В. С#/ Создание приложений для Windows.: Минск, Харвест., 2010.
6. Г. Шилдт. Полный справочник по С#. – М.: Издательский дом "Вильямс", 2008.
7. Э. Троелсен. С# и платформа .NET. Библиотека программиста. – СПб. : Питер, 2007.
8. Т.П. Караванова. Основи алгоритмізації та програмування. 750 задач з рекомендаціями та прикладами. – К.: Форум, 2002.
9. Э. Кингсли-Хьюджес, К. Кингсли-Хьюджес. С# 2005. Справочник программиста. – М.: ООО «ИД Вильямс», 2007.
10. Б. Керниган, Р. Пайк. Практика программирования. – СПб.: «Невский диалект», 2001.