

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

Кафедра механіки, автоматизації та інформаційних технологій



ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи

» _____ 2024 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК21 «Об'єктно-орієнтоване програмування»

(назва навчальної дисципліни)

Рівень вищої освіти: Перший (бакалаврський)

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: _____

Освітньо-професійна/наукова програма: «КОМП'ЮТЕРНІ НАУКИ»

ОНУ
2024

Робоча програма навчальної дисципліни «Об'єктно-орієнтоване програмування»
– Одеса: ОНУ, 2024. – 19 с.

Розробники:

Недева Ольга Анатоліївна, викладач;

Рачинська Алла Леонідівна, доцент;

Палій Катерина Сергіївна, викладач.

Робоча програма затверджена на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № 1 від “ 29 ” 08 2024 року

Завідувач кафедри _____ (Алла РАЧИНСЬКА)

(підпис)

Погоджено із гарантом ОПП/ОНП

«КОМП'ЮТЕРНІ НАУКИ»

_____ (Алла КАМЕНЄВА)

(підпис)

Схвалено навчально-методичною комісією (НМК)

з інформаційних технологій

Протокол № 1 від “ 30 ” 08 2024 року

Голова НМК _____ (Лариса МАРТИНОВИЧ)

(підпис)

Переглянуто та затверджено на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № _____ від “ _____ ” _____ 202_ року

Завідувач кафедри _____ (_____)

(підпис)

Переглянуто та затверджено на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № _____ від “ _____ ” _____ 202_ року

Завідувач кафедри _____ (_____)

(підпис)

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 годин - 240 змістових модулів - 3	Галузь знань <i>12 Інформаційні технології</i> (шифр і назва) Спеціальність <i>122 Комп'ютерні науки</i> (код і назва) Спеціалізації: <hr/> (назва) Рівень вищої освіти: <i>Перший (бакалаврський)</i>	Нормативна / за вибором (ВНЗ/студента)	
		Рік підготовки:	
		1-й	1-й
		Семестр	
		2-й	2-й
		Лекції	
		52 год.	12 год.
		Практичні, семінарські	
		год.	год.
		Лабораторні	
		68 год.	12 год.
		Самостійна робота	
		120 год.	216 год.
Форма підсумкового контролю: <i>іспит, курсова робота</i>			

2. Мета та завдання навчальної дисципліни

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#.

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) Інтегральна компетентність.

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

б) Загальні компетентності

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

в) Спеціальні (фахові, предметні) компетентності:

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

Програмні результати навчання:

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

ПР19. Володіти державною та іноземною мовами, вміти професійно спілкуватися у предметній області комп'ютерних наук як усно так і письмово.

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. Налаштування програми і обробка помилок.

Налаштування в Visual Studio. Налаштування в режимі зупинки. Покрокове виконання коду.

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

Складові частини визначення класу. Інкапсуляція даних. Властивості. Конструктор. Ключове слово this.

Тема 3. Об'єкти і класи.

Змінні типу посилання. Використання екземплярів класу в якості вхідних параметрів і значень, що повертаються. Методи класів, передача параметрів, повернення значень.

Тема 4. Часткові і статичні класи.

Ключове слово partial. Часткові методи. Використання модифікатора типу static. Визначення статичних класів.

Тема 5. Перевантаження операторів.

Ключове слово operator. Індексатори. Створення одновимірних індексаторів. Перевантаження індексаторів. Індексатори, що не потребують базовий масив. Багатовимірні індексатори.

Тема 6. Другий принцип ООП. Спадкування.

Похідний клас. Конструктори і успадкування. Приховування членів базового класу. Використання ключового слова sealed для запобігання успадкування. Другий принцип ООП: подробиці про спадкування. Визначення вкладених типів.

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

Віртуальні методи. Запечатування віртуальних членів. Абстрактні класи. Клас Object. Правила приведення до базового і похідного класу.

Змістовий модуль 2. Технологія WFA.

Тема 1. Технологія Windows Forms Application.

Основи візуального програмування інтерфейсу. Приклади простих застосунків. Керуючі елементи. Основні події.

Тема 2. Форма та стандартні компоненти.

Форма - основний компонент за стосунку технології Windows Forms Application. Керуючі елементи Button, Label, HScrollBar, VScrollBar, RadioButton та CheckBox. Меню. Пошук «зниклих» компонентів.

Тема 3. Контейнери.

Компоненти GroupBox та Panel . Батьки компонентів – Parent. Додаток з декількома формами. Приклад програми з модальною формою. Володіння вікнами. Загальні відомості про фокус. Компонент Timer. Функції-члени форми.

Тема 4. Події клавіатури та миші.

Події клавіатури KeyDown, KeyPress та KeyUp. Параметр типу KeyEventArgs. Обробка введення з клавіатури на рівні форми. Події миші MouseUp, MouseDown, Click, MouseClick і MouseMove. Параметр типу MouseEventArgs.

Тема 5. Вікно повідомлень.

Методу MessageBox.Show() класу MessageBox. Приклад програми з модальними формами заставки і запиту пароля.

Тема 6. Створення і використання динамічних компонент.

Властивість Controls. Методи Add() та AddRange(). Масиви динамічних компонент. Підключення подієвих функцій до динамічних компонентів. Створення нескінченного меню. Компонентний NumericUpDown. Компонент ToolTip.

Тема 7. Використання діалогових вікон.

Клас CommonDialog. Класи OpenFileDialog та SaveFileDialog. Діалогові вікна FontDialog та ColorDialog. Клас FolderBrowserDialog.

Тема 8. Керуючі елементи ListBox, ComboBox та PictureBox.

Вікна зі списком ListBox. Комбіноване вікно ComboBox. Клас PictureBox.

Тема 9. Створення бібліотек.

Збирання DLL. Приклад створення власної бібліотеки класу обертального руху. Підключення посилань до проекту. Тестування бібліотеки.

Тема 10. Побудова графіків.

Бібліотека ZedGraph.dll. Компонент ZedGraphControl. Використання рефлексії.

Тема 11. Технологія Drag&Drop.

Події DragDrop, DragEnter, DragLeave та DragOver. Тип DragEventArgs. Подієва функція GiveFeedback. Приклад створення застосування з використанням технології Drag&Drop.

Тема 12. Делегати і події.

Інкапсулювання методу за допомогою делегату. Багатоадресна передача. Події. Рекомендації щодо обробки подій у середовищі .NET Framework. Використання вбудованого делегата EventHandler.

Тема 13. Створення керуючих елементів.

Бібліотека керуючих елементів Windows. Додавання нових властивостей. Додавання нових обробників подій. Додавання власного оброблювача подій. Тестування доданих подій.

Тема 14. Елемент управління DataGridView.

Проектування компонента. Джерело даних доступне під час розробки. Відсутність джерела даних у дизайн-таймі. Готове джерело даних, що підключається під час виконання. Відсутність джерела даних під час виконання. Додавання нових рядків. Значення осередків.

Змістовий модуль 3. Узагальнення знань.

Тема 1. Серіалізація.

Серіалізація та десеріалізація. Приклади застосування серіалізацієї. Атрибут Serializable. Серіалізація та десеріалізація об'єктів із використанням SoapFormatter. Серіалізація та десеріалізація об'єктів за допомогою BinaryFormatter. Серіалізація та десеріалізація об'єктів з використанням XmlSerializer. Точність типів серед форматорів.

Роль графів об'єктів. Серіалізація та десеріалізація графа об'єктів за допомогою BinaryFormatter. Серіалізація та десеріалізація графа об'єктів із використанням SoapFormatter. Серіалізація та десеріалізація графа об'єктів з використанням XmlSerializer.

Серіалізація колекцій об'єктів.

Тема 2. UML – діаграми.

Діаграми класів. Відносини на рівні класів. Успадкування. Імплементация. Відносини на рівні об'єктів. Асоціація. Агрегація. Композиція.

4. Структура навчальної дисципліни

Назви тем	Кількість годин									
	Очна 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. Налаштування програми і обробка помилок	8	2			2	8	0,5			4
Тема 2. Перший принцип ООП. Визначення класів в C #	7	2		2	3	7	0,5		1	4
Тема 3. Об'єкти і класи	9	2		2	3	9	1		1	4
Тема 4. Часткові і статичні класи	5	2			3	5	0,5			5
Тема 5. Перевантаження операторів	8	2		2	2	8	0,5		0,5	5
Тема 6. Другий принцип ООП. Спадкування	7	2		3	3	7	1		1	5
Тема 7. Третій принцип ООП. Підтримка поліморфізму в C#	10	4		3	2	10	1		1	5
Тема 8. Інтерфейси.	4	2			2	4	0,5			5
Тема 9. Обробка виняткових ситуацій.	4	2			2	4	0,5			5
Разом за змістовим модулем 1	62	20		12	22	62	6		4,5	42
Змістовий модуль 2. Технологія WFA.										
Тема 1. Технологія Windows Forms Application.	13	2		4	5	13	0,5		0,5	9
Тема 2. Форма та стандартні компоненти.	12	2		4	5	12	0,5		0,75	9
Тема 3. Контейнери.	10	2		4	3	10			0,5	9
Тема 4. Події клавіатури та миші.	10	2		4	3	10	0,5		0,75	9

Тема 5. Вікно повідомлень	10	2		4	3	10			0,5	9
Тема 6. Створення і використання динамічних компонент	13	2		6	4	13	1		0,75	9
Тема 7. Використання діалогових вікон.	12	2		4	4	12	0,5		0,5	9
Тема 8. Керуючі елементи ListBox, ComboBox та PictureBox	10	2		4	3	10	0,5		0,5	9
Тема 9. Створення бібліотек.	10	2		4	3	10			0,5	9
Тема 10. Побудова графіків.	14	2		6	4	14			0,75	9
Тема 11. Технологія Drag & Drop.	16	2		6	6	16	1		1	10
Тема 12. Делегати і події	10	2		6	4	10	1		0,5	10
Тема 13. Створення керуючих елементів	11	2			5	11	0,5			10
Тема 14. Елемент управління DataGridView	13	2			6	13				10
Разом за змістовим модулем 2	164	28		56	58	164	6		7,5	130
Змістовий модуль 3. Узагальнення знань										
Тема 1. Серіалізація	8	2			6	8				8
Тема 2. UML – діаграми	6	2			4	6				6
Разом за змістовим модулем 3	14	4		0	10	14	0		0	14
Курсова робота	30				30	30				30
Усього годин	240	52		68	120	240	12		12	216

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	
		очна	заочна
1	Масиви масивів	5	1

2	Списки	5	1
3	Створення власного класу	5	1
4	Перевантаження операторів	5	1
5	Масиви і класи	6	1
6	Спадкування та поліморфізм	6	1
7	Вивчення принципів візуального програмування	7	1
8	Робота з текстовою інформацією	7	1
9	Динамічні компоненти	7	1
10	Обробка інформації, що зберігається в файлах	7	1
11	Технологія Drag & Drop	8	2
	Разом	68	12

8. Самостійна робота

№ з/п	Назва теми/ види завдань	Кількість годин	
		очна	заочна
1	Змістовий модуль 1. Принципи ООП	22	42
2	Змістовий модуль 2. Технологія WFA.	58	130
3	Змістовий модуль 3. Узагальнення знань	10	14
4	Курсова робота	30	30
	Разом	120	216

Теми курсових робіт.

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 рядам) 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 неможливо 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»

$$KI = \frac{a - b}{x}$$

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 – зафарбованою 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 предмета 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 кількома 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 клітинок (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 послідовних клітин, 1 - групу з однієї єдиної зафарбованою клітини). У чорно-білих кросвордах ми завжди зафарбовуємо клітинку чорним кольором, в кольорових - зафарбовуємо клітинку 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 виявляються різними, вони знову сховаються, а якщо в обох відкритих клітках буде одна і та ж картинка, то ці клітини залишаться 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»

У грі використовується стандартний набір – 28 кісток. Інша назва гри доміно п'ятірочка – `Muggins Domino`. Гра `Muggins` близький родич таких ігор як `All Fives` і `Sniff`, визнається багатьма як одна з найкращих доміно ігор. Грається за правилами звичайного доміно (`Draw Dominoes`), за винятком того, що мета гри - не просто вийти, але й виставляти на кінцях кістки, що дають у сумі 5 (або п'ять кратних). Після перемішування костяшок доміно, кожен гравець піднімає одну кістку для визначення, хто ходитиме першим, після чого кістки повертаються на стіл і знову перемішуються ("жеребкування"). Далі кожен гравець бере потрібну кількість кісток, яка залежить від кількості гравців:

2 гравці - по 7 кісток

3 гравці - по 5 кісток

4 гравці - по 5 кісток

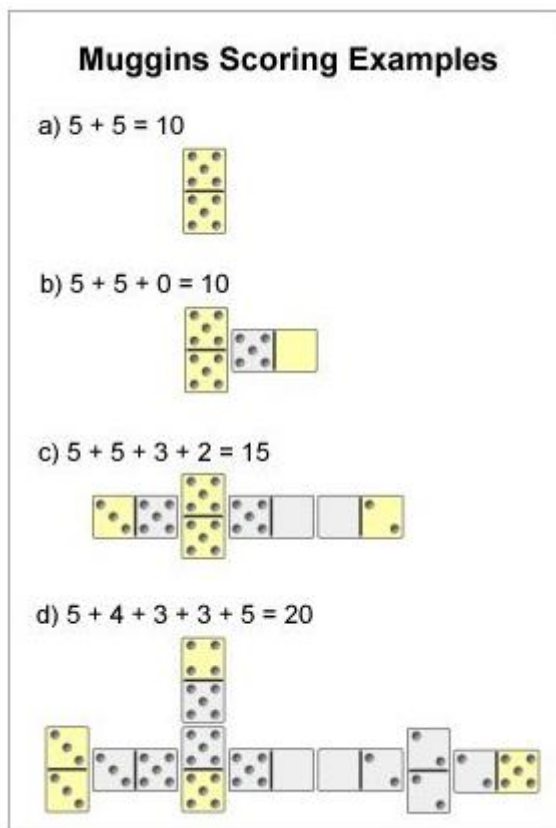
Кості, що залишили, складають «базар» (`boneyard` або `stock`).

Першим на початку партії ходить гравець, визначений жеребом (найменша кістка). Перший хід у здаванні – за загальними правилами. Ходити можна будь-якою кісткою, не обов'язково із дубля. Перший виставлений дубль, йдеться про перший хід чи ні, стає підставою хреста - кістки можна виставляти по всіх чотирьох краях. Усі наступні дублі є звичайними дублями.

Гра йде за годинниковою стрілкою. Якщо гравець не може зробити хід, він бере кістки з базару, поки він не знайде потрібну кістку. Якщо кістку не знайдено, гравець пропускає хід.

Мета гри полягає в тому, щоб набрати певну кількість очок (100, 250, 500). Очки нараховуються гравцеві, коли при виставленні своєї кістки сума відкритих на кінцях значень стає рівною 5 або кратній п'яти (5, 10, 15, 20 і т.д.).

У будь-який момент гри може бути відкрито 2, 3 або 4 кінці (без хреста – лише два). У дубля, що утворює хрест, при підрахунку, обліковуються всі чотири кінці (після його закриття з двох сторін). У звичайного дубля враховується його повна сума (6:6 – 12 очок).



Приклади:

- Якщо перша виставлена кістка 5-5, гравець отримує 10 балів. На даний момент усі сторони дубля 5-5 доступні для гри.
- Якщо друга кістка 5-0, гравець отримує 10 балів. На даний момент є три сторони по 5, а також порожньо.
- Якщо третя кістка 3-5 на дубль 5-5, сума становитиме 3 (3+0) і гравець не отримує балів. Якщо після виставлення 3-5 наступний гравець ставить 0-2, сума складає 5 (3+2) і гравець, який виставив 0-2, отримує 5 балів. Верхня та нижня частина хреста 5-5 залишаються доступними для гри, як і відкриті 3 та 2. У версії, реалізованій на Гамблері, після закриття двох сторін хреста кістка хреста не вважається. Лише кістки на закінченнях.
- Через чотири ходи 5-4 була поставлена на вершині 5-5, дубль 2-2 був поставлений до 0-2, 2-5 була поставлена до 2-2, дубль 3-3 був поставлений до 3-5. Сума 15 балів (4+3+3+5). Зараз відкрито 4 кінця: внизу хреста 5-5, 3-3, 4, та 5.

Дубль 2-2 та 3-3 є звичайними дублями – верхні та нижні кінці не доступні для гри.

Здача закінчується, коли один із гравців виставляє всі свої кістки, або коли гра блокується («риба»). Переможець (при «рибі» той, хто має найменшу кількість очок) отримує призові бали: очки на руках опонентів підсумовуються, округлюються до кратного 5 (3 до 5) і діляться на 5. Отримані бали додаються переможцю здачі. При «рибі» із загальної суми очок віднімаються очки у руці переможця. Наприклад, якщо гравець, що переміг, має 3 очки в руці, а троє його противників 5, 11, і 13, то загальна різниця $26 (5 + 11 + 13 - 3)$ - округляється до 25 і ділиться на 5. Переможець отримує 5 додаткових балів. Незалежно від перемоги, всі гравці зберігають свої бали, які вони отримали в грі.

Варіанти:

- У дубля, що утворює хрест, при підсумовуванні враховуються лише зіграні сторони: у прикладі "C" сума буде 5 (3+2), у прикладі "D" сума буде 15 (3+3+4+5).
- Без хреста.
- Набрані очки переможених не діляться на 5 і після округлення повністю записуються переможцю, при «рибі» очки переможця враховуються повністю (без урахування очок переможця).
- "Всі трійки" (All Threes) - в основному така ж гра, за винятком обліку кратності 3, а не 5.

9. Методи навчання

Словесні: лекції з аналізом конкретних прикладів, консультації.

Наочні: ілюстрація матеріалу у вигляді мультимедійних презентацій.

Практичні: лабораторні роботи; виконання індивідуальних завдань та індивідуальних контрольних работ.

10. Форми контролю і методи оцінювання

Методи поточного/періодичного контролю: оцінювання виконання лабораторних робіт та індивідуальних контрольних завдань.

Підсумковий контроль: Іспит. Під час підсумкового контролю студент повинен зробити індивідуальне технічне завдання.

Критерії оцінювання на поточному та підсумковому контролі:

Відмінно: додаток розроблено повністю згідно технічного завдання. На підсумковому контролі за указаний часовий термін.

Добре: додаток має неповний функціонал користувача, але вирішує основні задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

Задовільно: виконане візуальне проектування додатку, частково вирішено основні задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

Незадовільно: візуальне проектування застосування виконане частково, не вирішено ні однієї задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

11. Питання для підсумкового контролю

Під час проведення підсумкового контролю студент повинен розробити додаток згідно з індивідуальним технічним завданням, яке визначається білетом. Результатом виконання технічного завдання є Windows Forms Application.

12. Розподіл балів, які отримують студенти

Поточний та періодичний контроль											Підсумковий контроль (іспит)	Сума балів
Змістовий модуль 1						Змістовий модуль 2						
ЛР1	ЛР2	ЛР3	ЛР4	ЛР5	ЛР6	ЛР7	ЛР8	ЛР9	ЛР10	ЛР11		
5	5	5	5	6	6	8	8	9	9	9	15	100
контрольна робота -10												
курсова робота - 100												100

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 вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13. Навчально-методичне забезпечення

1. Електронний конспект лекцій за дисципліною «Об'єктно - орієнтоване програмування»:

<https://dspace.onu.edu.ua/items/b9d116ef-bada-4e13-95bf-5f785d8eebb9>

2. Лабораторні роботи за дисципліною «Об'єктно - орієнтоване програмування»:

https://drive.google.com/drive/folders/1TL8L18PiyMSuIRSr2MxIohyS5MyD_YBz?usp=drive_link

14. Рекомендована література

Основна

1. Бублик В.В. Об'єктно-орієнтоване програмування: підручник. Київ: ІТ- книга, 2015. 624 с.
2. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посіб. Львів: Видавництво Львівської політехніки, 2012. 624 с.
3. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно-орієнтованого програмування C#: навч. посіб. Київ: Державний університет телекомунікацій, 2018. 190 с.
4. Коноваленко І.В. Програмування мовою C# 6.0: навч. посіб. Тернопіль: ТНТУ. 2016. 227 с.
5. Бурлаков А.А. Об'єктно-орієнтований аналіз і проектування: метод. реком. Хмельницький: ХНУ, 2017. 136 с.
6. Daniel Solis. Illustrated C# 2012. Berkeley: APress, 2012. 732 p.
7. Standard ECMA-334. C# Language Specification. 4th Edition. Ecma International. June 2006 URL: https://www.ecma-international.org/wp-content/uploads/ECMA-334_4th_edition_june_2006.pdf
8. C# Reference, сайт розробників MSDN. URL: <https://code.visualstudio.com/docs/languages/csharp> (дата звернення 11.02.2023)

Додаткова

1. Щедрина О.І. Алгоритмізація та програмування процедур обробки інформації : навч. посіб. Київ: КНЕУ, 2001. 240 с.
2. Караванова Т.П. Основи алгоритмізації та програмування. 750 задач з рекомендаціями та прикладами: посіб. Київ: Форум, 2002. 287 с.
3. International standard ISO/IEC 23270:2006. Information technology – Programming languages. C#. Second edition. ISO/IEC. 2006.
4. The C# Programming Language / A. Hejlsberg et al. Boston: Addison-Wesley Professional, 2010. 864 p.
5. Albahari J., Albahari B. C# 7.0 in a Nutshell. Sebastopol: O'Reilly Media, 2017. 1090 p.
6. Schildt H. C# 4.0. The Complete Reference. New York: McGraw Hill Education, 2017. 984 p.
7. Sharp J. Microsoft Visual C# Step by Step. London: Pearson Education, 2018. 878p.
8. Troelsen A., Pro C# 7. With .NET and .NET Core. Berkeley: Apress, 2017. 1372 p.
9. Robert C. Martin, Clean Code/ A handbook of agile software craftsmanship, 2009, 464p.