

## ПРАКТИЧНА РОБОТА 2

### Інкапсуляція та взаємодія класів у Java-додатках

#### Система оцінювання

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Завдання на практичну роботу	3,8*
3.	<i>Здача звіту</i>	0,2
	<b>Всього</b>	<b>5</b>

\* – діє бонусна система

- 0,2 бала Напишіть клас Circle, який містить дані про радіус та колір круга (за замовчуванням 1.0 та «red»). Також клас повинен мати два перевантажені конструктори (безаргументний та з параметром-радіусом) і геттери для отримання радіусу та площі. Створіть об'єкт та виведіть дані щодо радіусу та площі круга.
- 0,2 бала Напишіть клас Author, який містить дані про ім'я, електронну пошту та статтю автора книги. Конструктора за замовчуванням не буде. Додайте в клас геттери/сеттери для даних, проте для імені та статі сеттерів не створюйте (ці дані мають бути незмінними). Метод toString() має виводити дані про автора в рядок. Створіть двох авторів.
- 0,2 бала (Масиви) Невелика авіакомпанія потребує систему автоматизації резервування місць у літаку на 10 місць. Ваша програма має пропонувати наступні альтернативи:  
Натисніть 1 для першого класу (місця 1-5)  
Натисніть 2 для економного класу (місця 6-10).  
Потім Ваш застосунок повинен відображати посадкові місця, вказуючи на замовлене місце та його клас.  
Використовуйте 1D-масив Boolean-значень для представлення посадкових місць у літаку. Ваш застосунок має ніколи не давати резервувати вже зарезервовані місця. Коли місця з економного класу закінчились, програма повинна запропонувати квиток першого класу. Якщо таке не підходить, програма виведе повідомлення «Наступний виліт через 3 години».
- 0,2 бала (Випадкові числа) Напишіть клас, який випадковим чином генерує автомобільні номери. Вважати, що автомобільний номер складається з 4 цифр, які знаходяться між двома парами латинських великих літер.
- 0,2 бала (Рядки та масиви) 13-цифровий міжнародний стандартний номер книги (International Standard Book Number) – це унікальний код, який ідентифікує книгу, що продається. Остання цифра називається перевіркою цифрою (check digit) і використовується для перевірки на помилки. Для обчислення цього значення

- а. кожен з 12 перших цифр по черзі множать на 1 або 3,
- б. додають добутки та ділять результат за модулем 10. Перевірочна цифра отримується відніманням від 10 отриманого результату.

Наприклад, для ISBN 9789609319614

$$(a) (9 \cdot 1 + 7 \cdot 3 + 8 \cdot 1 + 9 \cdot 3 + 6 \cdot 1 + 0 \cdot 3 + 9 \cdot 1 + 3 \cdot 3 + 1 \cdot 1 + 9 \cdot 3 + 6 \cdot 1 + 1 \cdot 3) = 126$$

$$(b) \text{The check digit} = 10 - (126 \% 10) = 10 - 6 = 4$$

Напишіть клас, в якому зчитується 13-цифровий ISBN та перевіряється останню цифру на валідність цього номеру.

6. *0,3 бала* Простий алгоритм шифрування даних пропонує записувати алфавіт в одному рядку, а в іншому – ті ж букви, але в іншому порядку. Другий рядок називається криптографічним ключем.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
d	y	p	r	i	a	j	u	h	t	q	w	e	s	f	o	v	c	n	b	l	x	m	k	z	g

Кожна з букв початкового тексту замінюється на відповідну букву ключа.

Напишіть програму, яка буде зчитувати рядок до 100 символів та замінитиме маленькі букви вхідного рядка на символи ключа. Потім програма повинна дешифрувати отриманий рядок та відновити початкове повідомлення.

7. *0,2 бала* Напишіть клас, який приймає на вхід дві матриці та перемножає їх, виводячи результат. Якщо матриці перемножити неможливо, потрібно вивести повідомлення про це.

8. *0,3 бала* Напишіть клас, який знаходить площу чотирикутника, утвореного чотирма містами України: Києвом, Львовом, Одесою, Харковом. Програма має отримувати GPS координати (довготу та широту) міст, знаходити відстані між ними за формулою

$$d = R \cdot \arccos(\sin x_1 \sin x_2 + \cos x_1 \cos x_2 \cos(y_1 - y_2)),$$

де  $R$  – радіус Землі (в середньому 6371,01км), а потім обчислювати площу чотирикутника як суму площ двох трикутників (кожна за формулою Герона), отриманих з'єднанням двох несусідніх міст.

Рекомендується окремо виділити метод для знаходження відстані між двома населеними пунктами, а також виокремити метод для обчислення площі прямокутника.

9. 0,2 бала Спроектуйте абстрактний клас GeometricObject, який відповідає UML-діаграмі.

Курсивом виділені абстрактні методи, які будуть реалізовуватись у задачі 9.

GeometricObject
-color: String -filled: boolean -dateCreated: java.util.Date
#GeometricObject() #GeometricObject(color: string, filled: boolean) +getColor(): String +setColor(color: String): void +isFilled(): boolean +setFilled(filled: boolean): void +getDateCreated(): java.util.Date +toString(): String +getArea(): double +getPerimeter(): double

10. 0,2 бала Спроектуйте клас Triangle, який розширяє клас GeometricObject. Цей клас має містити:

- Три поля, що відповідають сторонам трикутника (рівні 1.0 за замовчуванням)
- Безаргументний конструктор, що створює трикутник
- Конструктор, що створює трикутник із заданими сторонами
- Методи доступу (геттери/сеттери) до всіх трьох полів
- Метод getArea(), який повертає площу трикутника
- Метод getPerimeter(), який повертає периметр трикутника
- Метод toString(), який повертає опис трикутника

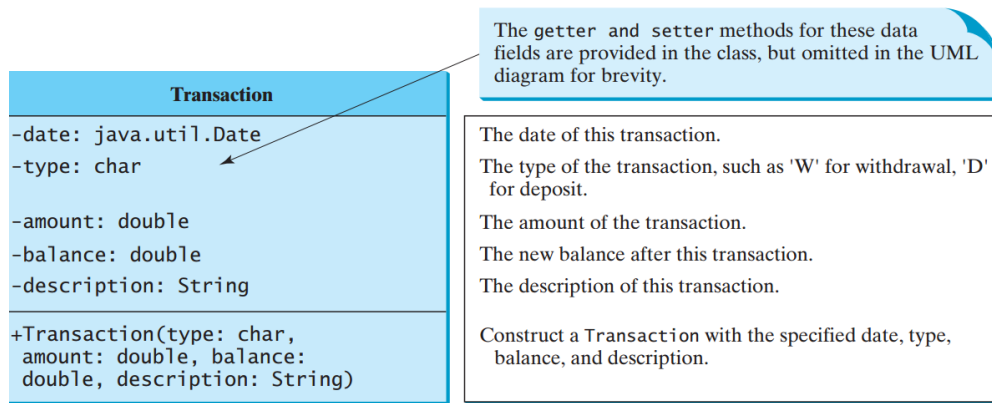
11. 0,3 бала Спроектуйте та реалізуйте клас Account, який буде описувати банківський рахунок та містити:

- Приватне цілочисельне поле id рахунку (за замовчуванням 0)
- Приватне дробове поле balance (за замовчуванням 0)
- Приватне дробове поле annualInterestRate, яке зберігає поточну відсоткову ставку (за замовчуванням 0). Припускаємо, що ставки для всіх рахунків однакові.
- Приватне поле dateCreated типу Date, яке зберігає дату відкриття рахунку.
- Безаргументний конструктор, який створює рахунок за замовчуванням.
- Конструктор, що відкриває рахунок із заданими id та стартовим балансом.
- Геттери/сеттери (accessor і mutator методи) для id, balance та annualInterestRate.
- Геттер для dateCreated.
- Метод getMonthlyInterest(), який повертає щомісячну процентну ставку для поточного балансу (не відсоток, а суму грошей).
- Метод withdraw(), який знімає з рахунку певну кількість коштів.
- Метод deposit() виконує внесок на рахунок.

Створіть рахунок з ID=1122, балансом 2000 грн. і відсотковою ставкою 18,5%. Спробуйте зняти 250 грн з рахунку, а потім покласти на нього 300 грн.

12. 0,3 бала На базі класу задачі 10 спроектуйте та реалізуйте його нову версію, що включатиме також:

- Нове поле name типу String, яке зберігатиме ПІБ власника рахунку
- Новий конструктор, який створює рахунок відповідно до name, id, balance.
- Нове поле transactions типу ArrayList, яке зберігає транзакції з рахунком. Кожна транзакція описується класом Transaction, діаграма якого зображена на рисунку



- Змініть методи `withdraw()` та `deposit()` з урахуванням спискового масиву транзакцій.

Протестуйте програму, створивши рахунок з відсотковою ставкою 14%, балансом 1000 грн, `id=1122`, а ім'я власника – Ваше. Зробіть внески по 50, 100 та 200 грн, а потім зніміть з рахунку 40, 20 та 10 грн. Виведіть повний звіт по рахунку: ім'я власника, процентну ставку, баланс та всі транзакції.

13. *0,3 бала* Напишіть простий консольний калькулятор, класи якого містять обробники для наступних виключень:

- `NumberFormatException`**, якщо хоча б одне введенне з клавіатури значення операнду не є числовим. Програма перед виходом має проінформувати користувача, що операнд має невірний тип.
- `InputMismatchException`** Оператор додавання має знаходити суму двох цілочисельних операндів, введених користувачем. Якщо ввід некоректний, програма має знову запросити користувача ввести дані.
- `ArrayIndexOutOfBoundsException`** Окремо напишіть метод, який створює масив зі 100 випадкових цілих чисел, пропонує користувачеві ввести індекс елементу масиву і виводить відповідне значення. Якщо заданий індекс виходить за рамки, слід вивести повідомлення **Out of Bounds**.

14. *0,3 бала* Припустимо, слід розробити меню ресторану. Обмежимося супами. Створіть клас Страви, який містить статичний вкладений клас Суп (це буде helper-клас у вигляді перерахування з доступних супів, їх типів та кількості зірок, що визначають їх рейтинг). Нехай в меню присутні томатний суп (тип – овочевий, 5 зірок), суп з курчам (тип – м'ясний, рейтинг ще не виставлено) та креветковий суп (тип – суп з морепродуктів, 3 зірки). Приклад виводу такого меню:

```

c:\_00Concepts>java RecipeTest
We have TOMATO soup in our list, which is a vegetable soup and is 5 star rated
We have CHICKEN soup in our list, which is a meat soup and is not rated yet
We have PRAWN soup in our list, which is a seafood soup and is 3 star rated
c:\_00Concepts>
  
```

15. *0,4 бала* (Гра в зонк) Напишіть клас-заготовку, який дозволяє грати виконувати один хід у гру «зонк». За правилами гри може бути більше двох гравців. Для гри потрібно 6 шестигранних кубиків, які будуть кидатись одночасно. Доступні наступні комбінації:

Категория	Название	Очки	Пример	Примечания
	Единицы	100 за каждую		Комбинация из трех единиц дает 1000 очков. Однако в различных вариациях правил четвертая единица дает как 100, так и 1000 очков.
	Пятерки	50 за каждую		Комбинация из трех пятерок дает 500 очков. Однако в различных вариациях правил четвертая пятерка дает как 50, так и 500 очков.
<b>Три одинаковых</b>				
	Единицы	1000		
	Двойки	200		
	Тройки	300		
	Четверки	400		
	Пятерки	500		
	Шестерки	600		
<b>Специальные</b>				
	Четвертая/ пятая/ шестая кость	Приносит столько же, сколько три предыдущих кости	 +  = 300 + 300	Это касается и пятой, и шестой кости. В некоторых вариантах правил четвертая кость не обладает этим свойством.
	Три пары	750 и призовой бросок		Если выпали 4 кости одного достоинства и 2 другого (например,  ), эта комбинация 750 очков не принесет. Кроме того, комбинацию "три пары" можно получить только за один бросок.
	Шесть разных	1500 (1000) и призовой бросок		В других вариантах правил эта сумма может быть несгораемой или за эту комбинацию дается 1500 очков.

Виграє той, у кого буде більше очок за один хід.

Користувачу потрібно ввести кількість гравців та їх ім'я і прізвище. Далі обирається один з гравців (виводиться запрошення до гри), генерується 6 випадкових чисел (від 1 до 6), а потім гравець обирає доступні комбінації. Програма має перевірити, чи присутня така комбінація для кинутих костей та додати бали гравцю за її наявності. Інакше вивести повідомлення про те, що комбінація недоступна. Загальні очки порівнюються. Якщо існує більше одного переможця, їм надаються по одному додатковому ходу.