

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА

В. А. Бородін

Методичні рекомендації з курсу «Мова
програмування Java»

xxxxx

“zzzz”

2024

yghjhjihjookop

Розділ 1

Простий клас Java. Командний рядок. Методи та функції

Завдання для аудиторної роботи:

- 1.1 Створіть клас, що містить неініціалізовані поля типів `int`, `char` та `Рядок`, і надрукуйте їх значення, щоб переконатися, що Java виконує ініціалізацію за замовчуванням. Напишіть програму "hello, world". Скомпілюйте програму з `javac` і запустіть її за допомогою `java`. Якщо ви використовуєте інше середовище розробки, ніж `JDK`, дізнайтеся, як компілювати та запускати програми в цьому середовищі.
- 1.2 Напишіть програму, яка друкує три аргументи, взяті з командного рядка. Знайти суму аргументів командного рядка та порахувати введені дійсні числа.
- 1.3 Обчисліть наступні математичні вирази та виведіть результати:
 - $1.2 + 31$;
 - $45 \cdot 54 - 11$;
 - $\frac{15}{4}$;
 - $\frac{15.0}{4}$;
 - $67\%5$;
 - $\frac{2 \cdot 45.1 + 3.2}{2}$
- 1.4 Ввести дійсне число від 0 до 10000 та вивести його 8 ступінь з точністю до 20 знаків до десяткової коми та 4 значками після десяткової коми.
- 1.5 Під функціями в цьому розділі розуміються статичні методи класу. Напишіть функцію, яка за найменшу кількість арифметичних операцій, обчислює значення многочлена для введеного з клавіатури значення:
 - $y = x^4 + 2x^2 + 1$;
 - $y = x^4 + x^3 + x^2 + x + 1$;
 - $y = x^5 + 5x^4 + 10x^3 + 10x^2 + 5x + 1$;
 - $y = x^9 + x^3 + 1$;
 - $y = 16x^4 + 8x^3 + 4x^2 + 2x + 1$;
 - $y = x^5 + x^3 + x$.
- 1.6 Трикутник заданий довжинами своїх сторін. Знайти периметр та площу цього трикутника. Перевірте для значень сторін $a = 3$, $b = c = 3.5 + 3 \cdot 2^{-111}$. Трикутник вводиться координатами своїх вершин, які вводяться так: впершому рядку через пробіл два дійсних числа --- координати точки А, пропускається рядок, в третьому рядку через пробіл два дійсних числа --- координати Б, пропускається рядок, через пробіл – координати точки С. Підрахувати площу трикутника. (Вказівка: напишіть функції підрахунку довжини відрізка та функції обчислення площі трикутника за довжинами сторін)

Завдання для самостійної роботи:

- 1.1 Вивести в косоль підказку «Введіть прізвище», введіть його з консолі та виведіть повідомлення «Привіт, ***» де замість зірочок — введене прізвище.
- 1.2 Ініціалізувати два довільні рядки та вивести їх в одному рядку та поставивши між ними кому та пробіл, а перед та після три окличних знаки, як вприкладі: !!! Hello , World! !!!
- 1.3 Введіть два цілих числа та виведіть кількість їх розрядів (розв'яжить це за допомогою рядків та використовуючі логарифм). Обчисліть середнє гармонічне цих чисел та виведіть його з точністю до двох знаків після коми.

1.4 Ввести цілі числа як аргументи командного рядку, підрахувати їх добуток та вивести результат на консоль.

1.5 Обчислити результати наступних виразів та вивести на екран написану українською мовою «Результат дорівнює:»:

- $2 + 3$;
- $4.5 \cdot 56$;
- $\frac{2}{3.0}$

1.6 Дійсне число записано в рядку, при цьому перед ним може стояти будь-яка послідовність з пробілів та символів `*`. Ввести його та виведіть значення його кубу.

1.7 Ініціалізуйте наступні числа як дійсні та подвійні дійсні: 10^{-4} ; $2.33E5$; π ; e ; $\sqrt{5}$; $\ln 100$.

1.8 Задайте в програмі довільні 5 цілих та 5 дійсних чисел. Вивести на екран таблицю з цих значень у вигляді, сліdkуючі за "красою" виводу:

```
x | 1 | 2 | 3 | 4 | 5 |
- | - | - | - | - | - |
y | 3.0 | 1.0 | 5.0 | 4.0 | 2.1 |
```

1.9 Обчислити силу притягання F в науковому (експоненційному) форматі між двома тілами, що мають маси m_1, m_2 на відстані r . Вказівка. Шукана сила визначається за формулою $F = \gamma \frac{m_1 \cdot m_2}{r^2}$, де $\gamma = 6.673 \cdot 10^{-11} \text{ Н} \cdot \text{м}^2 / \text{кг}^2$. Всі потрібні змінні присвоюються всередині програми. Результат вивести в окремому рядку вигляду « $F = ***$ », де замість зірок представлення в науковому (експоненційному) вигляді.

1.10 Ввести, користуючись лише однією функцією вводу, ціле число записане в шістнадцятковому вигляді та вивести його зменшене на одиницю в шістнадцятковому та десятковому вигляді.

1.11 Знайти об'єм тора з внутрішнім радіусом r і зовнішнім радіусом R . Знайти об'єм тора з внутрішнім радіусом r і зовнішнім радіусом R .

1.12 Ввести дійсне число x та обчислити значення функції тригонометричного косинуса для нього.

1.13 В трикутнику відомо довжини всіх сторін. Обчислити довжини його: медіан, бісектрис, висот.

1.14 Знайти дійсні корені квадратного рівняння з коефіцієнтами a, b, c якщо відомо, що обидва корені в ньому існують. Перевірте ваш розв'язок на коефіцієнтах рівняння $a = 3, b = 100, c = 2$.

1.15 Напишіть функцію $Rosenbrock2d(x, y) = 100(x^2 - y)^2 + (x - 1)^2$ та перевірте її результат на довільних трьох парах дійсних чисел.

1.16 Ввести два цілих числа то обчислити їх добуток (так щоб він був коректний навіть при максимально великих цілих числах) та частку як дійсне число з максимально можливою кількістю цифр після десяткової крапки.

1.17 Ввести два дійсних числа записаних через пробіли в одному рядку та обчислити значення їх середнього арифметичного та середнього гармонічного. Результат вивести в науковому та десятковому представленні.

1.18 Ввести дійсне число x та підрахуйте без та за допомогою математичних функцій її цілу та дробову частину, найменше ціле число, що більше x та найбільше ціле, що менше x , а також його округлене значення. Перевірте результат роботи для від'ємного числа.

1.19 Обчислити відстань від точки (x_0, y_0) до:

- заданої точки (x, y) ;
- заданої прямої $(ax + by + c = 0)$;
- точки перетину прямих $(x + by + c = 0)$ і $(ax + y + c = 0)$, де $(ab \neq 1)$.

1.20 Введіть два цілих числа та виведіть кількість їх розрядів (розв'яжіть це за допомогою рядків та використовуючі логарифм). Обчисліть середнє гармонічне цих чисел та виведіть його з точністю до двох знаків після коми.

Розділ 2

Оператори та типи даних Java. Умовні конструкції. Цикли.

Завдання для аудиторної роботи:

- 2.1 Напишіть програму, яка ініціалізує два цілих числа, одне задане з першим бітом рівним нулю, а інше, з одиницею у першому біті (підказка: Найпростіше використовувати шістнадцяткову систему для цього або бінарну). Візьміть ці два значення та виконайте всі можливі операції з ними, використовуючи побітові оператори, і виведіть результати за допомогою Integer.toString().
- 2.2 Дано три натуральних числа: 32-бітне число, 8-бітний номер біта та булеве значення біта. Необхідно поміняти в першому числі біт з заданим номером на передане значення та вивести отримане після змін біта число в десятковій, шістнадцятковій та двійковій формі. Наприклад, якщо Ви вводите «12 1 1», програма має вивести «13 0xD 1101»; і якщо Ви вводите «13 1 0» програма має вивести «12 0xC 1100»
- 2.3 Вивести усі перестановки з N елементів у лексикографічному порядку. Для N=3: 123, 132, 213, 231, 312, 321
Вивести усі комбінації потужності K з множини розміру N . Для пари чисел (3; 2) це: 1 2, 1 3, 3 1, 2 3, 3 1, 3 2.
- 2.4 Напишіть дві функції для розрахунку факторіалу числа: одна функція рахує факторіал за допомогою циклу, інша за допомогою рекурсії. Виведіть на консоль результати обчислень обох функцій; якщо число не є натуральним, виведіть повідомлення «Число не натуральне!».
- 2.5 Таксі Рамануджана. С. Рамануджан був індійським математиком, який прославився своєю інтуїцією щодо чисел. Коли одного разу англійський математик Г. Х. Харді приїхав до нього в госпіталь, Харді зауважив, що номер його таксі - 1729, досить нудне число. На що Рамануджан відповів: "Ні, Харді! Це дуже цікаве число. Це найменше число, яке можна виразити як суму двох кубів двома різними способами".
Перевірте це твердження, написавши програму Ramanujan.java, яка приймає цілий аргумент командного рядка n і друкує всі цілі числа, менші або рівні n , які можна виразити як суму двох кубів двома різними способами - знайдіть різні додатні цілі числа a, b, c і d такі, що $a^3 + b^3 = c^3 + d^3$. Тепер номер 87539319 здається досить нудним номером. Визначте, чому це не так.

Завдання для самостійної роботи:

- 2.1 Напишіть метод, який відображає значення char у двійковому вигляді. Продемонструйте це, використовуючи декілька різних символів.
- 2.2 Ввести натуральне однобайтове число n (однобайтове число) і вивести число, отримане в результаті циклічного зсуву числа n на один розряд вліво, тобто старший біт посунутий в позицію молодшого, а всі інші біти зсуваються на один розряд вліво. Наприклад, якщо введено 130, відповіддю буде 5.
- 2.3 Визначити, скільки разів зустрічається дана послідовність 0 та 1 (задана однобайтовим числом) в двійковому поданні цілого додатнього числа (так, в двійковому поданні 11010111 послідовність 11 зустрічається 3 рази, а послідовність 10 - 2 рази).
- 2.4 У сучасному світі будь-які дані в пам'яті комп'ютера і при їх передачі по каналах зв'язку представлені у вигляді послідовності байтів. У різних системах і протоколах байти прийнято упорядковувати по-різному: від старшого до молодшого (big-endian) або від молодшого до старшого (little-endian). Наприклад, маємо число $2017_{10} = 11111100001_2$, що в протоколі Big-endian записується як $2017_{10} = 00000111 \ 11100001_2$, а в протоколі Little-endian як $2017_{10} = 11100001 \ 00000111_2$
В блокчейн різницю між кодуванням чисел можна побачити: Bitcoin використовує little-endian, Ethereum використовує big-endian.

Напишіть програму, яка перетворює число до формату з little-endian з урахуванням того для гіпотетичної обчислювальної системи, де використовуються байти складаються тільки з 7 біт. На вхід подається ціле число N ($1 < N < 2^{28} - 1$).

Необхідно вивести послідовність з 28 символів 0 або 1 (біти) - представлення вхідного числа в форматі little-endian.

- 2.5 Напишіть функцію, яка приймає у якості першого аргументу число яке представляє довжину, а якості другого --- величину виміру як рядок вигляду "мм", "см", "км" і т.п. Функція повинна правильно виводити довжину в метрах.
- 2.6 Число українською. Напишіть програму для зчитування в цілому числі командного рядка між - 999,999,999 та 999,999,999 та виведить літерний еквівалент. Ось вичерпний перелік слів, які має використовувати ваша програма: мінус, нуль, один, два, три, чотири, п'ять, шість, сім, вісім, дев'ять, десять, одинадцять, дванадцять, тринадцять, чотирнадцять, п'ятнадцять, шістнадцять, сімнадцять, вісімнадцять, дев'ятнадцять, двадцять, тридцять, сорок, п'ятдесят, шістдесят, сімдесят, вісімдесят, дев'яносто, сто, тисяча, мільйон.
- 2.7 Ввести число від 1 до 12, друге число від 1 до 31 та третє число від 2000 до 2100. Перевірити, чи це коректна дата. Вивести цю дату з назвою місяця українською.
- 2.8 Напишіть програму FivePerLine.java, яка за допомогою одного циклу for та одного оператора if друкує цілі числа від 1000 до 2000 з п'ятьма цілими числами на рядок.
- 2.9 У п'ятницю 13-го. Яка максимальна кількість днів поспіль, у які не буває п'ятниці 13-го? Підказка: Григоріанський календар повторюється кожні 400 років (146097 днів), тому вам потрібно турбуватися лише про 400-річний інтервал. Результат: 426 (наприклад, з 13.08.1999 по 13.10.2000).
- 2.10 Напишіть програму, яка приймає цілий аргумент командного рядка n і друкує цілі числа від 1 до 999 з n чисел на рядок. Зробіть цілі числа вирівняними, надрукуювши правильну кількість пробілів перед числом (наприклад, три для 1-9, два для 10-99 і один для 100-999).
- 2.11 Число "вампіра" має парну кількість цифр і формується шляхом множення пари чисел, що містять половину цифр результату. Цифри беруться з вихідного номера в довільній послідовності. Пари кінцевих нулів не допускаються.
Приклади: $1260 = 21 \cdot 60$, $1827 = 21 \cdot 87$, $2187 = 27 \cdot 81$.
Напишіть програму, яка знайде всі 4-значні числа вампірів.
- 2.12 Ввести послідовність наступним чином: користувачу виводиться напис " $a[**]=$ ", де замість $**$ стоїть номер числа, що вводиться. Тобто там виводиться напис " $a[0]=$ ", і після знаку рівності користувач вводить число, " $a[1]=$ ", і користувач вводить друге число, і так далі, доки користувач не введе число 0. Після цього потрібно вивести суму введених чисел.
- 2.13 Напишіть програму RollLoadedDie.java, яка друкує результат кидання кубика так, що ймовірність отримання 1, 2, 3, 4 або 5 дорівнює $1/8$, а ймовірність отримання 6 - $3/8$.
- 2.14 Припустимо, що a , b і c - випадкові числа, рівномірно розподілені між 0 і 1. Яка ймовірність того, що a , b і c утворюють деякий трикутник?
Підказка: вони складуть трикутник тоді і тільки тоді, коли сума кожних двох значень більша за третю. Використайте метод Монте-Карло.
- 2.15 У ігровому шоу «Давайте укладемо угоду» учаснику показують три двері.
За одними дверима цінний приз, за двома іншими - нічого. Після того, як учасник обирає двері, ведучий відкриває одну з інших двох дверей (звичайно, ніколи не розкриваючи приз).
Після цього учаснику надається можливість перейти до інших невідчинених дверей.
Чи повинен це зробити учасник? Інтуїтивно може здатися, що двері першого вибору конкурсанта та інші невідкриті двері однаково ймовірно міститимуть приз, тому не буде стимулу змінювати думку.
Напишіть програму MonteHall.java, щоб перевірити цю інтуїцію шляхом моделювання. Ваша програма повинна взяти цілий аргумент командного рядка n , пограти в гру n разів, використовуючи кожну з двох стратегій (перемикання чи не перемикання) і надрукувати шанс на успіх для кожної стратегії.

Розділ 3

Масиви та ініціалізація класів

Завдання для аудиторної роботи:

- 3.1 Створіть масив об'єктів String та ініціалізуйте його. Роздрукуйте масив за допомогою циклу for () в один рядок через кому та роздрукуйте в кожний рядок в новому рядку з вказанням індексу масиву.
- 3.2 Кількість слів і рядків. Напишіть програму, щоб вона читала текст з консолі та друкувала кількість символів, слів та рядків у тексті.
- 3.3 У масивах монотонний відрізок - це кілька послідовних елементів які зростають, або спадають. Наприклад: масив -1 3 2 5 6 7 3 2 1 1 1 має монотонні відрізки: -1 3, 2 5 6 7, 7 3 2 1, 1, 1.

Завдання полягає у виведенні всіх монотонних відрізків максимальної довжини.

Для прикладу вище: такий відрізок всього один довжиною 4: 7 3 2 1, а в прикладі 1 2 1 2 1 таких відрізків 4: 1 2, 2 1, 1 2, 2 1.

Необхідно враховувати обидва порядки!

На вхід програма отримує в першому рядку довжину масиву, далі йдуть елементи масиву.

Виведіть всі відрізки, кожен з нового рядка.

- 3.4 Ввести з консолі матрицю $a[n][n]$ для натуральних n . Написати методи задання значення елементів матриці в інтервалі значень від $-n$ до n за допомогою датчика випадкових чисел та за допомогою консолі та виводу матриці.

Завдання для самостійної роботи:

- 3.1 Напишіть програму MaxMin.java що дозволяє користувачу ввести скільки завгодно цілих чисел в новому рядку, доки той не введе число 0, та виводить найбільше та найменше значення серед цих чисел.
- 3.2 Напишіть програму, що вводять цілі числа та виводить число яке зустрічається підряд найбільшу кількість разів та скільки саме разів (якщо таких чисел декілька - то виводиться найбільше з таких чисел) в заданому нижче форматі. Наприклад, в послідовності 1 2 2 1 5 1 1 7 7 7 1 1 це «4 підряд чисел 7»
- 3.3 Напишіть програму Closest.java, що вводять з командного рядку числа x, y, z (координату точки A) та читає з консолі масив чисел (x_i, y_i, z_i) , і виводить ту з нових точок, що найближча до (x, y, z) .
- 3.4 Знайти серед масиву чисел таке число, в якому кількість цифр мінімальна. Якщо їх декілька, виведіть всі з них.
- 3.5 Серед чисел знайти число-паліндром. Якщо таких чисел більше, ніж одне, то виведіть друге з них.
- 3.6 Вводяться двовимірні координати та маси послідовності об'єктів (кожен об'єкт в окремому рядку, дані - через кому).
Напишіть програму для обчислення їх центру мас чи центроїда.
Центроїд - це середнє положення n об'єктів, зважене за масою. Програма зчитує послідовності позицій і мас (x_i, y_i, m_i) з консолі і роздруковує їх центр мас (x, y, m) .

- 3.7 Булева матриця має властивість парності, коли кожен рядок і кожен стовпець мають парну суму. Це простий тип коду для виправлення помилок, тому що якщо один біт пошкоджено під час передачі (біт перевертається від 0 до 1 або від 1 до 0), його можна виявити та виправити. Напишіть програму ParityCheck.java, яка приймає ціле число N як вхідний сигнал командного рядка і зчитує в булевій матриці N -по- N зі стандартного входу, та виводить, що:

- матриця має властивість парності, або
- вказує, який єдиний пошкоджений біт (i, j) можна перевернути, щоб відновити властивість парності, або
- вказує, що матриця була пошкоджена (для відновлення властивості парності потрібно змінити більше двох бітів).

Використовуйте якомога менше внутрішньої пам'яті. Підказка: вам навіть не потрібно зберігати матрицю!

3.8 Ввести з консолі матрицю $a[n][n]$ для натуральних n . Впорядкувати рядки (стовпці) матриці в порядку зростання значень елементів рядка(стовпчика).

3.9 Ввести з консолі матрицю $a[n][n]$ для натуральних n . Повернути матрицю на 90 (180, 270) градусів проти годинникової стрілки.

3.10 Ввести з консолі матрицю $a[n][n]$ для натуральних n . Знайти максимальний елемент(ти) в матриці і видалити з матриці всі рядки і стовпці, що містять його.

3.11 Ввести з консолі матрицю $a[n][n]$ для натуральних n . Після цього:

- Обчислити норму матриці.
- Обчислити визначник матриці.
- Ущільнити матрицю, видаляючи з неї рядки і стовпці, заповнені нулями.

3.12 Внесення змін. Напишіть програму, яка зчитує в командному рядку ціле число N (кількість копійок) і друкує найкращий спосіб (найменша кількість монет) для нарахування даної суми за допомогою монет (лише 1, 25, 10 та 50 копійок). Наприклад, якщо $N = 73$, надрукуйте «50 коп 10 коп * 2 1 коп * 3». Підказка: використовуйте жадібний алгоритм.

3.13 Двадцять питань. Напишіть програму QuestionsTwenty.java, яка виводить 20 запитань: користувач задумує число від 1 до мільйона, а комп'ютер здогадується, що це за число, запитуючи користувача, більше, менше чи дорівнює воно якомусь числу. Використовуйте двійковий пошук, щоб переконатися, що комп'ютеру потрібно не більше 20 відгадок.

3.14 Рюкзак. Дано: кількість предметів, ваги предметів та вагу, яку може витримати рюкзак.

Необхідно вибрати кілька предметів і покласти їх в рюкзак так, щоб не перевищити його максимальну вагу. Поліноміального (оптимального) рішення немає. Завдання вирішується рекурсивно повним перебором. Використовуйте деревоподібну рекурсію, де в одному випадку ми беремо предмет і йдемо до наступного, а в іншому не беремо і йдемо до наступного. Необхідно вивести максимальну сумарну вагу предметів, які можна покласти у рюкзак. Спочатку водяться два цілих числа: N - кількість предметів, W - максимальна вага рюкзака. Далі N цілих чисел - ваги предметів. Потрібно підрахувати та вивести ціле число – максимальну сумарну вагу всіх предметів, які можна покласти у рюкзак. Наприклад, якщо Ви введете «5 12 1 2 3 4 5», то програма має вивести «12». Інший приклад: за вхідних даних даних «5 17 1 3 18 3 15» отримуємо відповідь «16»

3.15 Коды UPC. Універсальний код продукту (UPC) - це 12 -значний код, який однозначно визначає продукт. Найменш значуща цифра d_{12} (крайня права) - це контрольна цифра, яка визначається однозначно, зробивши такий вираз кратним 10:

$$(d_1 + d_3 + d_5 + d_7 + d_9 + d_{11}) + 3(d_2 + d_4 + d_6 + d_8 + d_{10} + d_{12})$$

Наприклад, контрольна цифра, що відповідає 0-48500-00102 (Tropicana Pure Premium Orange Juice), дорівнює 8, оскільки $(8 + 0 + 0 + 0 + 5 + 4) + 3(2 + 1 + 0 + 0 + 8 + 0) = 50$ і 50 кратно 10.

Напишіть програму, яка зчитує 11 -значне ціле число з параметра командного рядка, обчислює контрольну цифру та друкує повний UPC.

Підказка: використовуйте змінну типу long для збереження 11 -значного числа.

Розділ 4

Створення власних класів. Наслідування.

Завдання для аудиторної роботи:

- 4.1 Створіть клас у пакеті. Створіть екземпляр свого класу поза цим пакетом. Покажіть, що захищені методи мають доступ до пакетів, але не є загальнодоступними.
- 4.2 Визначити клас Чотирикутник на площині. Визначити площу і периметр чотирикутника. Створити масив / список / множину об'єктів і підрахуйте кількість чотирикутників різного типу (квадрат, прямокутник, ромб, довільний). Визначити для кожної групи найбільший і найменший за площею (периметром) об'єкт.
- 4.3 Створіть клас під назвою ConnectionManager, який керує фіксованим масивом об'єктів Connection. Клієнт-програміст не повинен мати можливість явно створювати об'єкти Connection, але може отримати їх лише статичним методом у ConnectionManager. Коли у ConnectionManager закінчуються об'єкти, він завершує роботу.
- 4.4 Створити об'єкт класу Текст, використовуючи класи Речення, Слово. Методи: доповнити текст, вивести на консоль текст, заголовок тексту.
- 4.5 Створити консольний додаток, який задовольняє наступні вимоги:
 - а) Використовує можливості ООП: класи, спадкування, поліморфізм, інкапсуляція.
 - б) Спадкування застосовується тільки тоді, коли це має сенс.
 - в) При кодуванні повинні бути використані угоди по оформленню коду java code convention.
 - г) Класи повинні бути грамотно розкладені по пакетах.
 - д) Консольне меню повинно бути мінімальним.Квіткаря. Визначити ієрархію квітів. Створити кілька об'єктів-квіток. Зібрати букет (використовуючи аксесуари) з визначенням його вартості. Провести сортування квітів в букеті на основі рівня свіжості. Знайти квітку в букеті, що відповідає заданому діапазону довжини стебла.
- 4.6 Реалізувати інтерфейси, також використовувати успадкування та поліморфізм interface: Видання - abstract class - Книга - class - Довідник і Енциклопедія.
- 4.7 Створити клас Payment з внутрішнім класом, за допомогою об'єктів якого можна сформувати покупку з декількох товарів.

Завдання для самостійної роботи:

Для завдань створити консольний додаток, який задовольняє вимоги з завдання 4.5 аудиторної роботи. Визначити для кожного класу конструктори та метод toString(). Для кожного члену класу Type - публічні методи setType(), getType(). Визначити додатково методи в класі, що створює масив об'єктів. Задати критерій вибору даних і вивести ці дані на консоль.

Спадкування застосовувати тільки в тих завданнях, в яких це логічно обґрунтовано. Аргументувати приналежність класу кожного створюваного методу і коректно перевизначити для кожного класу методи equals (), hashCode (), toString ().

У кожному класі, що містить дані, має бути оголошено декілька конструкторів.

- 4.1 Student: id, Прізвище, Ім'я, По батькові, Дата народження, адресу, телефон, Факультет, Курс, Група. Створити масив об'єктів. Вивести: а) список студентів заданого факультету; б) списки студентів для кожного факультету і курсу; в) список студентів, які народилися після заданого року; г) список навчальної групи.
- 4.2 Product: id, Найменування, UPC, Виробник, Ціна, Термін зберігання, Кількість.

Створити масив об'єктів. Вивести: а) список товарів для заданого найменування; б) список товарів для заданого найменування, ціна яких не перевищує задану; с) список товарів, термін зберігання яких більше заданого.

- 4.3 Bus : Прізвище та ініціали водія, Номер автобуса, Номер маршруту, Марка, Рік початку експлуатації, Пробіг.

Створити масив об'єктів. Вивести: а) список автобусів для заданого номера маршруту; б) список автобусів, які експлуатуються більше заданого терміну; с) список автобусів, пробіг у яких більше заданої відстані.

- 4.4 Визначити клас Квадратне рівняння. Реалізувати методи для пошуку коренів, екстремумів, а також інтервалів убуття / зростання. Створити масив / список / множину об'єктів і визначити найбільші і найменші значення коренів.

- 4.5 Точка на площині (в просторі) та в часі. Задати рух точки у певному напрямку. Створити методи по знаходженню швидкості та прискорення точки. Перевірити для двох точок можливість перетину траєкторій. Визначити відстань між двома точками в заданий момент часу.

- 4.6 Визначити клас Поліном з коефіцієнтами типу КомплекснеЧисло. Оголосити масив / список із m поліномів і визначить суму поліномів масиву.

- 4.7 Створити об'єкт класу Держава, використовуючи класи Область, Район, Місто. Методи: вивести на консоль столицю, кількість областей, площу, обласні центри.

- 4.8 Створити об'єкт класу ОдновимірнийМасив, використовуючи класи Масив, Елемент. Методи: створити, вивести на консоль, виконати операції (скласти, відняти, помножити).

- 4.9 Камені. Визначити ієрархію дорогоцінних і напівдорогоцінних каменів. Відібрати камені для намиста. Підрахувати загальну вагу (в каратах) і вартість. Провести сортування каменів намиста на базі цінності. Знайти камені в намисті, відповідні заданому діапазону параметрів прозорості.

- 4.10 Реалізувати інтерфейси, також використовувати успадкування та поліморфізм interface: Корабель - abstract class - Військовий Корабель - class - Авіаносець.

- 4.11 Створити клас Department з внутрішнім класом, за допомогою об'єктів якого можна зберігати інформацію про всі посадах відділу і про всіх співробітників, коли-небудь займали конкретну посаду.

Розділ 5

Рядки. Регулярні вирази. Текстові файли

Завдання для аудиторної роботи:

5.1 Надрукувати заданий рядок:

- виключивши з нього всі цифри і подвоївши знаки '+' та '-';
- виключивши з нього всі знаки '+', безпосередньо за якими знаходиться цифра;
- виключивши з нього всі літери 'в', безпосередньо перед якими знаходиться літера 'с';
- замінивши в ньому всі пари 'ph' на літеру 'f';
- виключивши з нього всі зайві пропуски, тобто з кількох, що йдуть підряд, залишити один.

5.2 Напишіть програму, яка в перед усіма словами "small" та "large" додасть слово "very".

5.3 Уявіть, що ви працюєте у великій компанії, де використовується модульна архітектура. Ваш колега написав модуль із якоюсь логікою (ви не знаєте) і передає у вашу програму якісь дані. Ви пишете функцію яка зчитує дві змінних типу string, а повертає число типу int, яке потрібно отримати за допомогою додавання цих рядків. Але не все так просто, адже ваш колега не пише на Java, і він злий через те, що за неї платять більше. Тому він вирішив пожартувати з вас і підсунув вам каверзу. Він придумав вставляти сміття в рядки перед тим, як викликати вашу функцію. Тому вам потрібно прибрати з них сміття і конвертувати в число. Під сміттям маються на увазі зайві символи та спеціалізовані знаки. Наприклад: «%^80 hhhhh20&&&nd» конвертується в «100»

5.4 Дано текстовий файл. Групи символів, що відокремлені пропусками (одним або кількома) і не містять пропусків усередині, називатимемо словами. Скласти підпрограми для:

- визначення кількості слів у файлі;
- вилучення з файла зайвих пропусків і всіх слів, що складаються з однієї літери.

Завдання для самостійної роботи:

5.1 Напишіть програму, що записує текст у зворотному порядку.

5.2 Напишіть програму, яка перевіряє, чи певне слово міститься в даному тексті. Розгляньте випадки, коли текст визначається в самій програмі і вводиться з командного рядка при запуску програми на виконання. Розгляньте випадки залежно/незалежно від регістру.

5.3 Напишіть програму, яка підраховує число символів, слів, знаків пунктуації та речень у тексті. Розгляньте випадки, коли текст визначається в самій програмі і вводиться з командного рядка при запуску програми на виконання.

5.4 Перевірити, чи складається рядок з:

- 2 симетричних підрядків;
- n симетричних підрядків.

5.5 Визначте, чи може даний рядок бути валідною адресою публічної електронної пошти.

Для логіну валідні: символи латиниці в малому та великому регістрі, символи: точка, нижнє підкреслення, тире, а також цифри. Довжина логіну не може перевищувати 32 символи.

Електронна пошта складається з імені користувача, символу @ та домену.

Валідним доменним ім'ям вважатимемо рядок виду «ім'я компанії, точка, країна розміщення» (univ.ua, toyota.jp і т.д.)

Для найменування компанії використовуються символи з того ж набору, що і для логіну, довжина так само не повинна перевищувати 32 символи.

Країна розміщення: 2 символи в латиниці в малому регістрі, а також рядки com, net, org.

5.6 В заданий рядок входять тільки цифри та літери. Перевірте це. Визначити, чи задовольняє він наступній властивості:

- рядок починається з деякої ненульової цифри, за якою знаходяться тільки літери і їх кількість дорівнює числовому значенню цієї цифри;
- рядок містить (крім літер) тільки одну цифру, причому її числове значення дорівнює довжині рядка;
- рядок складається тільки з цифр, причому їх числові значення складають арифметичну прогресію (наприклад, 3 5 7 9, 8 5 2, 2).

5.7 Дано рядок. Групи символів, що відокремлені пропусками (одним або кількома) і не містять пропусків усередині, називатимемо словами. Скласти підпрограми для:

- знаходження найдовшого слова;
- визначення кількості слів
- вилучення з рядку зайвих пропусків і всіх слів, що складаються з однієї літери;
- вилучення всіх пропусків на початок рядків, у кінці рядків і між словами (крім одного);
- вставки пропусків до рядків рівномірно між словами так, щоб довжина всіх рядків (якщо в них більше 1 слова) була 80 символів і кількість пропусків між словами в одному рядку відрізнялась не більш ніж на 1 (вважати, що рядки файлу мають не більш ніж 80 символів).

5.8 Визначити процедуру, яка переписує компоненти текстового файлу F до файлу G, вставляючи до початку кожного рядка один символ пропуску. Порядок компонент не має змінюватися.

5.9 Інверсія рядка A - це рядок B, записаний тими ж символами у зворотньому порядку. Інверсія порожнього рядка за означенням – порожній рядок. Побудувати інверсію рядка. Результат в цьому самому рядку.

5.10 Написати програму, яка виконує зсув по ключі (ключ задається) тільки для малих латинських та українських літер. Наприклад: вхідні дані - anz – рядок, 2 – ключ. Результат: cpb.

5.11 Визначити функцію, що визначає кількість рядків текстового файлу, що:

- починаються із заданого символу;
- закінчуються заданим символом;
- починаються й закінчуються одним і тим самим символом;
- складаються з однакових символів.

5.12 У текстовому файлі кожний рядок містить кілька натуральних чисел, які розділяються пропусками. Числа визначають вигляд геометричної фігури (номер) та її розміри. Прийнято такі домовленості: відрізок прямої задається координатами своїх кінців і має номер 1; прямокутник задається координатами верхнього лівого й нижнього правого кутів і має номер 2; коло задається координатами центра й радіусом і має номер 3. Визначити процедури обчислення:

- відрізка з найбільшою довжиною;
- прямокутника з найбільшим периметром;
- кола з найменшою площею.

5.13 Створіть клас Студент - який містить інформацію про студента (ПІБ, курс, номер заліковки, кількість зданих предметів та відповідний масив оцінок). Реалізуйте метод, що дозволяє створювати файл з даного масиву студентів та метод для додавання запису про студента в масив. Реалізуйте метод, що знаходить студента з даним прізвищем у даному файлі та виводить його в інший файл та метод знаходження прізвища студента з найгіршим середнім балом.

5.14 У двох файлах міститься таблиця футбольного турніру, у першому – записано назви команд; у другому – результати матчів, що зберігаються у записах типу T_Match

```
{
int n1, n2; int b1, b2;

} T_Match;
```

Тут у класі T_Match поля n1, n2 – номери першої і другої команд (тобто номери назв команд у файлі команд); b1, b2 – кількість м'ячів, забитих першою та другою командами, відповідно. Кожній команді за перемогу нараховується 3 очки, за нічию – 1, за поразку – 0. Із двох команд, які мають однакову кількість очок, першою вважається та, що має кращу різницю забитих і пропущених м'ячів. Знайти команду, яка є лідером.

Указівка. Описати методи створення файлів команд і матчів, додавання результату матчу, визначення лідера.

Розділ 6

Бінарні файли. Серіалізація.

Завдання для аудиторної роботи:

- 6.1 Визначити процедуру, яка переписує до текстового файла G усі рядки текстового файла F із заміною в них символа '0' на '1', і навпаки.
- 6.2 Прочитати з бінарного файлу дійсні значення (до кінця файлу), знайти їх суму та вивести на консоль.
- 6.3 Описати класи Факультет та Інститут (з полем – масивом факультетів). Створити об'єкти, здійснити їх бінарну серіалізацію й десеріалізацію.

Завдання для самостійної роботи:

- 6.1 Робота з файлом з дійсних чисел. Створити бінарний файл з випадкових цілих чисел.
 - Прочитати з бінарного файлу дійсні значення (до кінця файлу), знайти їх суму та вивести на консоль.
 - Прочитати з бінарного файлу дійсні значення (до кінця файлу), знайти добуток модулів ненульових елементів та вивести в інший бінарний файл.
- 6.2 Робота з файлом з цілих чисел. Створити бінарний файл з випадкових цілих чисел.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), знайти добуток парних елементів та вивести в інший бінарний файл.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), замінити від'ємні значення модулями, додатні нулями та вивести отримані значення в інший бінарний файл.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), замінити розділити парні елементи на 2, непарні – збільшити у 2 рази та вивести отримані значення в інший бінарний файл.
- 6.3 Дано файл, компоненти якого є дійсними числами. Скласти підпрограми для обчислення:
 - суми компонент файлу;
 - кількості від'ємних компонент файлу;
 - останньої компоненти файлу;
 - найбільшого зі значень компонент файлу;
 - найменшого зі значень компонент файлу з парними номерами.
- 6.4 Робота з файлом з цілих чисел. Створити бінарний файл з випадкових цілих чисел.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), знайти добуток парних елементів та вивести в інший бінарний файл.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), замінити від'ємні значення модулями, додатні нулями та вивести отримані значення в інший бінарний файл.
 - Прочитати з бінарного файлу цілі значення (до кінця файлу), замінити розділити парні елементи на 2, непарні – збільшити у 2 рази та вивести отримані значення в інший бінарний файл.
- 6.5 Дано файл F, компоненти якого є цілими числами. Файл містить рівне число додатних і від'ємних чисел. Використовуючи допоміжний файл H, переписати компоненти файла F до файла G так, щоб у файлі G:
 - не було двох сусідніх чисел одного знака;
 - спочатку йшли додатні, потім – від'ємні числа;
 - числа йшли таким чином: два додатних, два від'ємних тощо (припускається, що число компонент у файлі F ділиться на 4).
- 6.6 Серіалізація: Багаж пасажирів характеризується номером пасажирів, кількістю речей і їхньою загальною вагою. Дано файл пасажирів, який містить прізвища пасажирів і файл, що містить інформацію про багаж пасажирів (номер пасажирів – це номер запису у файлі пасажирів)

Скласти процедури для:

- а) знаходження пасажирів у багажі якого середня вага однієї речі відрізняється не більш ніж на 1 кг від загальної середньої ваги речей;
- б) визначення пасажирів, які мають більше двох речей, і пасажирів кількість речей у яких більша за середню кількість речей;
- в) видачі відомостей про пасажирів, кількість речей у багажі якого не менша, ніж у будь-якому іншому багажі, а вага речей – не більша, ніж будь-якому іншому багажі із цією самою кількістю речей.

6.7 Задано натуральне число n та файл F , компоненти якого є цілими числами. Побудувати файл G , записавши до нього найбільше значення перших n компонент файла F , потім – наступних n компонент тощо. Розглянути два випадки:

- кількість компонент файла ділиться на n ;
- кількість компонент файла не ділиться на n . Остання компонента файла G має дорівнювати найбільшій із компонент файла F , які утворюють останню (неповну) групу.

6.8 Дано файл, який містить відомості про іграшки: указано назву іграшки (напр., м'яч, лялька, конструктор тощо), її вартість у гривнях і вікові межі для дітей, яким іграшка призначається (напр., для дітей від двох до п'яти років). Скласти процедури: а) пошуку іграшок, вартість яких не перевищує 40 грн; б) пошуку іграшок, призначених дітям від трьох до семи років; в) пошуку іграшок, ціна яких відрізняється від ціни найдорожчої іграшки не більш ніж на 50 грн; г) найдорожчого конструктора; д) пошуку всіх кубиків; е) пошуку двох іграшок, сумарна вартість яких є мінімальною і не перевищує 20 грн; ж) пошуку конструкторів вартістю 22 грн, призначених дітям від п'яти до десяти років

6.9 Відомості про учня складаються з його імені, прізвища та назви класу (рік навчання та літери), в якому він вчиться. Дано файл, який містить відомості про учнів школи. Скласти підпрограми, які дозволяють:

- а) визначити, чи є в школі учні з однаковим прізвищем;
- б) визначити, чи є учні з однаковим прізвищем у паралельних класах;
- в) визначити, чи є учні з однаковим прізвищем у певному класі;
- г) відповісти на питання а)-в) стосовно учнів, у яких збігаються ім'я та прізвище;
- д) визначити, в яких класах налічується більше 35 учнів;
- е) зібрати у файл відомості про учнів 9-10-х класів, розташувавши
- є) спочатку відомості про учнів класу 9 а, потім – 9 б тощо;
- ж) отримати список учнів даного класу за зразками:

Прізвище Ім'я

Прізвище І.

І.Прізвище.

6.10 Створити клас для роботи з комплексними числами та написати клас, що дозволяє серіалізувати/десеріалізувати об'єкти комплексних чисел.

6.11 Серіалізація: Багаж пасажирів характеризується номером пасажирів, кількістю речей і їхньою загальною вагою. Дано файл пасажирів, який містить прізвища пасажирів, і файл, що містить інформацію

про їхній багаж (номер пасажира – це номер запису у файлі пасажирів). Скласти процедуру для знаходження пасажира, у багажі якого середня вага однієї речі відрізняється не більш ніж на 1 кг від загальної середньої ваги речей.

6.12 Нехай множина цілих чисел задана у файлі. Визначити:

- процедуру введення множини;
- процедуру виведення множини;
- процедуру доповнення множини;
- процедуру видалення елемента з множини;
- функцію, що дає відповідь, чи входить елемент до множини;
- функцію, що дає відповідь, чи порожня множина;

Розділ 7

Колекції

Завдання для аудиторної роботи:

- 7.1 Створити в стеку індексний масив для швидкого доступу до записів в бінарному файлі.
- 7.2 Ввести рядки з файлу, записати в список ArrayList. виконати сортування рядків, використовуючи метод sort () з класу Collections.
- 7.3 Напишіть програму, яка обчислює відносну частоту появи певних слів зі списку (які вводяться з командного рядочку) в даному тексті, незалежно від регістру.
- 7.4 У колі стоять N людей, пронумерованих від 1 до N . При веденні програм повинна використовувати клас ArrayList, а друга - LinkedList. Яка з двох програм працює швидше? Чому? Перевірте це.
- 7.5 На площині задано N точок. Вивести в файл опису всіх прямих, які проходять більш ніж через одну точку із заданих. Для кожної прямої вказати, через скільки точок вона проходить. використовувати клас HashMap.

Завдання для самостійної роботи:

- 7.1.ю Ввести рядки з файлу, записати в список. Вивести рядки в файл в зворотному порядку.
- 7.2. Ввести число, занести його цифри в стек. Вивести число, у якого цифри йдуть в зворотному порядку.
- 7.3. Занести вірші одного автора в список. провести сортування по зростанню довжин рядків.
- 7.4. Створити та ініціалізувати два стека, поміняти інформацію в них місцями.
- 7.5. Визначити множину на основі множини цілих чисел. Створити методи для визначення перетину і об'єднання множин.
- 7.6. Помножити два многочлена заданого ступеня, якщо коефіцієнти многочленів зберігаються в різних списках.
- 7.7. Списки (стеки, черги) $I(1..n)$ і $U(1..n)$ містять результати n вимірів струму і напруги на невідомому опорі R . Знайти наближене число R методом найменших квадратів.
- 7.8. З використанням множини виконати попарне підсумовування довільного кінцевого ряду чисел за такими правилами: на першому етапі підсумовуються попарно поруч стоять числа, на другому етапі підсумовуються результати першого етапу і т.д. до тих пір, поки не залишиться одне число.
- 7.9. Заданий список цілих чисел і число X . Не використовуючи допоміжних об'єктів і не змінюючи розміру елементів списку зробити так, щоб спочатку йшли числа, які не перевищують X , а потім числа, що більші за X .
- 7.10. У колі стоять N людей, пронумерованих від 1 до N . Вводиться число m - кількість слів у лічилці, яка починається з першого номера. Виведіть порядок, у якому люди будуть вибувати з кола. При веденні програм повинна використовувати клас ArrayList, а друга - LinkedList. Яка з двох програм працює швидше? Чому? Перевірте це
- 7.11. Реалізувати клас Graph, що представляє собою неорієнтований граф. У конструкторі класу передається кількість вершин в графі. Методи повинні підтримувати швидке додавання і видалення ребер.
- 7.12. У вхідному файлі розташовані два набору позитивних чисел; між наборами є негативне число. Побудувати два списки $C1$ і $C2$, елементи яких містять відповідно числа 1-го і 2-го набору таким чином,

щоб усередині одного списку числа були впорядковані по зростанню. Потім об'єднати списки C1 і C2 в один впорядкований список, змінюючи тільки значення полів референтного типу.

7.13. У вхідному файлі зберігаються дві розріджені матриці A і B. Побудувати циклічно пов'язані списки CA і CB, що містять ненульові елементи відповідно матриць A і B. Переглядаючи списки, обчислити:

а) суму $S = A + B$;

б) добуток $P = A * B$.

7.14. Дана матриця з цілих чисел. Знайти в ній прямокутну підматрицю, що складається з максимальної кількості однакових елементів. Використовувати клас Stack.

7.15. На площині задано N точок. Вивести в файл опису всіх прямих, які проходять більш ніж через одну точку із заданих. Для кожної прямої вказати, через скільки точок вона проходить. Використовувати клас HashMap.

7.16. На прямій гоночній трасі курсує N автомобілів, для кожного з яких відомі початкове положення і швидкість. Визначити, скільки відбудеться обгонів.

7.17. На листочку в клітинку намальоване коло. Вивести в файл опису всіх клітин, що цілком лежать всередині кола, в порядку зростання відстані від клітини до центру кола. Використовувати клас PriorityQueue.

7.18. На аркуші папері в клітинку зафарбована частина клітин. Виділити все різні фігури, які утворилися при цьому. Фігурою вважається набір зафарбованих клітин, досяжних один з одного при русі в чотирьох напрямках. Дві фігури є різними, якщо їх не можна поєднати поворотом на кут, кратний 90 градусам, і паралельним переносом. Використовуйте клас HashSet.

7.19. Один із способів шифрування даних, званий «подвійним шифруванням», полягає в тому, що вихідні дані за допомогою деякого перетворення послідовно шифруються на деякі два ключа K1 і K2. Алгоритм шифрування - просто додавання по модулю 2 слова до ключа. Розробити і реалізувати ефективний алгоритм, що дозволяє знаходити ключі K1 і K2 по заданому рядку і його зашифрованого варіанту.

Розділ 8

Потоки. Багатопоточність

Завдання для аудиторної роботи:

- 8.1 Один потік кожні t_1 одиниць часу надає повідомлення. Інший потік має обробити ці повідомлення. Обробка займає t_2 одиниць часу. Скласти програму, яка генерує та обробляє повідомлення. Повідомлення мають оброблятися у порядку їх надходження. Обробка повідомлення – це просто показ його на екрані. Використати чергу.
- 8.2 Call-центр має N операторів. Через випадковий час від t_1 до t_2 до call-центру надходять дзвінки клієнтів. Оператор може одночасно обслуговувати тільки одного клієнта. Час обслуговування клієнта займає від t_3 до t_4 одиниць часу. Якщо всі оператори зайняті, клієнт кладе трубку і передзвонює ще раз через деякий випадковий час від t_5 до t_6 . Промодельовати роботу call-центру. Один клієнт – це 1 потік.
- 8.3 Вільна каса. У ресторані швидкого обслуговування є кілька кас. Відвідувачі стоять в черзі в конкретну касу, але можуть перейти в іншу чергу при зменшенні або зникненні там черги. Промодельовати роботу такого ресторану.

Завдання для самостійної роботи:

- 8.1 Порт має N причалів. Через випадковий час від t_1 до t_2 до порту причалює новий корабель для розвантаження або завантаження контейнерів та займає один вільний причал. Якщо вільних причалів немає, корабель чекає поки якийсь причал не звільниться. Завантаження та розвантаження одного контейнеру займає t_3 та t_4 одиниць часу, відповідно. Кількість контейнерів, яку має завантажити або розвантажити окремий корабель визначається довільним значенням від 1 до m . Промодельовати роботу порту, якщо до нього має причалити n_1 кораблів для розвантаження та n_2 кораблів для завантаження контейнерів. Один корабель – це 1 потік.
- 8.2 Тунель. В горах існує два залізничних тунелю, за якими поїзди можуть рухатися в обох напрямках. По обох кінцях тунелю зібралося багато поїздів. Забезпечити безпечне проходження тунелів в обох напрямках. Поїзд можна перенаправити з одного тунелю в інший при перевищенні заданого часу очікування на проїзд.
- 8.3 Стоянка має N місць. Через випадковий час від t_1 до t_2 до стоянки приїздить новий автомобіль, займає довільне вільне місце та через випадковий час від t_3 до t_4 покидає стоянку. На одному місці може знаходитися тільки один автомобіль. Якщо всі місця зайняті, то автомобіль не стане чекати і поїде на іншу стоянку. Промодельовати роботу стоянки та обчислити кількість зароблених стоянкою грошей, якщо за одиницю часу на стоянці водій має заплатити m грошей. Один автомобіль – це 1 потік.
- 8.4 Готель має N номерів. Через випадковий час від t_1 до t_2 приходить новий клієнт та заселяється у один з номерів (якщо є вільний), у якому живе час від t_3 до t_4 . Якщо вільних номерів немає, то клієнт очікує на звільнення будь-якого номера. Промодельовати роботу готелю та розрахувати середній час очікування для заданої кількості клієнтів. Один клієнт – це 1 потік.
- 8.5 Аеропорт. Посадка / висадка пасажирів може здійснюватися через кінцеве число терміналів іноземним способом через кінцеве число трапів. Літаки бувають різної місткості і дальності польоту. Організувати функціонування аеропорту, якщо пунктів призначення 4-6, і зон дальності 2-3.
- 8.6 Біржа. На торгах брокери пропонують акції кількох фірм. На біржі здійснюються дії з купівлі-продажу акцій. Залежно від кількості проданих-куплених акцій їх ціна змінюється. Брокери пропонують до продажу деяку частину акцій. Від активності і зростання-падіння котирувань акцій змінюється індекс біржі. Біржа може призупинити торги при різкому падінні індексу.

- 8.7 Місце для паркування. Доступно декілька машиномісць. На одному місці може знаходитися тільки один автомобіль. Якщо всі місця зайняті, то автомобіль не стане чекати більше певного часу і поїде на іншу стоянку. Про моделюйте стоянку за допомогою генератора випадкових чисел.
- 8.8 Банк. Є банк з касирами, клієнтами та їх рахунками. Клієнт може знімати / поповнювати / переводити / оплачувати / обмінювати грошові кошти. Касир послідовно обслуговує клієнтів. Потік-спостерігач стежить, щоб в касах завжди були готівкові, при скупченні грошей більше певної суми, частина їх перекладається в сховище, при виснаженні запасів готівки відбувається поповнення зі сховища.
- 8.9 Аукціон. На торги виставляється кілька лотів. Учасники аукціону роблять заявки. Заявку можна коригувати в бік збільшення кілька разів за торги одного лота. Аукціон визначає переможця і переходить до наступного лоту. Учасник, що не заплатив за лот в заданий проміжок часу, відсторонюється на кілька лотів від торгів.
- 8.10 Створити багатопотоковості додаток з 1 параметром для нижченаведеного завдання.
Є такі потоки: кухар, офіціантка і відвідувачі, число відвідувачів задано параметром. Кожного відвідувача потрібно нагодувати обідом з 3 страв.
Кухар готує 1 блюдо, офіціантка бере його і відносить 1 відвідувачеві, потім (випадковим чином) кухар або готує 2 блюдо і воно відноситься офіціанткою 1 клієнту, або кухар готує 1 блюдо для 2 клієнта. Наступного разу кухар вже може приготувати або 1 блюдо для нового клієнта (якщо залишилися клієнти), або чергове блюдо, яке ще не отримував попередній клієнт. Але якщо, наприклад, 2 клієнта чекають другу страву, то першим його повинен отримати клієнт з меншим номером. Якщо клієнт не отримав ще другу страву, то він не може отримати третє блюдо. Вивести номер страви разом з ім'ям "кухар" або "офіціантка" або "клієнт" з номером клієнта в залежності від того, хто працює в даний момент часу. Робота потоків закінчується, коли всіх відвідувачів обслужили. Виведення може бути таким:

Блюдо1- кухар

Блюдо1-офіціантка

Блюдо1-клієнт1

Блюдо1- кухар

Блюдо1-офіціантка

Блюдо1-клієнт2

Блюдо2- кухар

Блюдо2-офіціантка

Блюдо2-клієнт1

Блюдо1- кухар

Блюдо1-офіціантка

Блюдо1-клієнт3

Блюдо3- кухар

Блюдо3-офіціантка

Блюдо3-клієнт1 ...

Розділ 9

Мережа

Завдання для аудиторної роботи:

- 10.1 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт читає з текстового файлу та відправляє на сервер рядки. Сервер до кожного такого рядка додає номер та повертає його клієнту. Клієнт зберігає рядки у новому файлі.
- 10.2 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт спочатку відправляє на сервер певний символ. Потім клієнт читає з текстового файлу та відправляє на сервер рядки. Сервер зберігає у новому текстовому файлі тільки ті рядки, які містять заданий символ, та відправляє клієнту повідомлення чи був збережений рядок.

Завдання для самостійної роботи:

- 10.1 Розробіть додаток, який підключається до веб-сервера і отримує поточний час. Виведіть цей час у консоль або графічному інтерфейсі користувача.
- 10.2 Напишіть програму, яка здійснює запит до веб-сайту, отримує вміст сторінки і аналізує її, знаходячи певні дані. Наприклад, можна отримати курси валют з фінансового сайту або новини з новинного порталу.
- 10.3 Створіть мережеву гру, де гравці можуть підключатися до сервера і взаємодіяти один з одним. За допомогою цього реалізуйте гру «Морський бой».
- 10.4 Розробіть додаток, який дозволяє користувачам обмінюватися файлами через мережу. Користувачі можуть завантажувати файли на сервер або взяти файли з сервера.
- 10.5 Напишіть програму, яка моніторить стан певних серверів або служб у вашій мережі. Програма може відправляти запити на сервери і перевіряти, чи вони відповідають, а потім відображати статус цих серверів у реальному часі.
- 10.6 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки, в яких записана послідовність цілих чисел. Сервер для кожного такого рядка знаходить максимальне та мінімальне число і повертає їх клієнту.
- 10.7 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки. Сервер для кожного такого рядка знаходить максимальне за довжиною слово і повертає його клієнту (якщо таких слів декілька, повертається перше з них). Словом можна вважати послідовність символів, що розділена пропусками.
- 10.8 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт спочатку відправляє на сервер шаблон регулярного виразу. Потім клієнт читає з текстового файлу та відправляє на сервер рядки. Сервер для кожного рядка перевіряє, чи міститься в даному рядку підрядок, що відповідає заданому шаблону, та надсилає відповідь клієнту.
- 10.9 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт відправляє на сервер рядки. Сервер для кожного такого рядка знаходить усі цифри, які входять у заданий рядок, і повертає їх клієнту.
- 10.10 Скласти програми сервера та клієнта для розв'язання такої задачі. Клієнт спочатку відправляє на сервер натуральне число. Потім клієнт відправляє на сервер рядки, в яких записана послідовність натуральних чисел. Сервер для кожного такого рядка знаходить числа, які є кратними заданому, і повертає їх кількість клієнту

Розділ 10

Інтерфейси та патерни програмування

Завдання для аудиторної роботи:

- 9.1 Розглянемо систему Безпечний комплекс, який складається з двох основних елементів: Центр управління та Зона безпеки. Центр управління відповідає за координацію та зберігання конфіденційної інформації, тоді як Зона безпеки - це місце зосередження персоналу та обладнання.

Створіть шаблон програмування Відвідувач (Visitor) для моделювання відвідування об'єктів безпеки конкурентами. Конкуренти можуть бути двох типів: Промислові шпигуни та Злочинці. Метою відвідування Промисловим шпигуном є збір та викрадення конфіденційної інформації, тоді як Злочинцям призначено знищення конфіденційних документів, персоналу та обладнання в безпечній зоні.

Для цього створіть інтерфейс Шпигун, який містить абстрактні методи для відвідування різних типів об'єктів безпеки. Класи Конфіденційний агент та Злочинець реалізують цей інтерфейс та виконують свої завдання відповідно до специфікації.

- 9.2 Створення білдера для конструювання замовлень в інтернет-магазині

Уявіть, що ви працюєте над розробкою функціоналу для інтернет-магазину. Одним із завдань є створення системи, яка дозволяє користувачам легко складати замовлення для покупки товарів. Вам потрібно використати паттерн побудови "білдер" для реалізації конструктора замовлень.

Вимоги:

- Створіть інтерфейс OrderBuilder, який визначає методи для додавання товарів до замовлення, встановлення адреси доставки та інших параметрів.
- Створіть клас Order, який буде представляти саме замовлення. У цьому класі повинен бути конструктор, що приймає OrderBuilder і використовує його для ініціалізації замовлення.
- Створіть клас-реалізацію DefaultOrderBuilder, який імплементує інтерфейс OrderBuilder. Цей клас повинен мати методи для додавання товарів, встановлення адреси доставки, а також будь-яких інших параметрів, які можуть бути потрібні для замовлення.
- Напишіть клас OrderBuilderDemo, де ви створите об'єкт Order за допомогою OrderBuilder і продемонструєте його використання

Завдання для самостійної роботи:

- 9.3 Система пошуку транспортних засобів

Уявіть, що ви розробляєте систему для пошуку різних типів транспортних засобів, таких як автомобілі, велосипеди та мотоцикли. Вам потрібно використати паттерн Фабричний метод для створення об'єктів цих транспортних засобів через їх спільний інтерфейс.

Вимоги:

- Створіть інтерфейс Transport, який описує методи, що повинні бути реалізовані всіма транспортними засобами, наприклад, move() для руху.
- Створіть класи, що реалізують інтерфейс Transport: Car, Bicycle та Motorcycle. Кожен з цих класів повинен реалізувати метод move(), який відображає рух конкретного транспортного засобу.

- Створіть абстрактний клас `TransportFactory`, який міститиме абстрактний метод `createTransport()`, що повертатиме об'єкт типу `Transport`.
- Створіть конкретні підкласи `TransportFactory` для кожного типу транспорту: `CarFactory`, `BicycleFactory` та `MotorcycleFactory`. Кожен з цих класів повинен реалізувати метод `createTransport()`, що повертатиме відповідний об'єкт транспортного засобу.
- Напишіть клас-демонстрацію `TransportDemo`, де ви створите об'єкти різних типів транспортних засобів за допомогою фабричних методів і продемонструєте їх рух.

9.4 Створення фабрик для створення електронних пристроїв

Уявіть, що ви розробляєте систему для створення електронних пристроїв, таких як смартфони та планшети. Вам потрібно використати паттерн Абстрактна фабрика для створення фабрик, які будуть виробляти різні моделі цих пристроїв через їх спільні інтерфейси.

Вимоги:

- Створіть інтерфейси `Smartphone` та `Tablet`, які описують методи, що повинні бути реалізовані всіма смартфонами та планшетами відповідно.
- Створіть класи, що реалізують інтерфейси `Smartphone` та `Tablet`: наприклад, `SamsungSmartphone`, `SamsungTablet`, `AppleSmartphone`, `AppleTablet` і т.д. Кожен з цих класів повинен реалізувати методи, визначені відповідним інтерфейсом.
- Створіть абстрактні фабрики `ElectronicDeviceFactory`, які містять абстрактні методи `createSmartphone()` та `createTablet()`, що повертають об'єкти типу `Smartphone` та `Tablet`.
- Створіть конкретні фабрики, які розширюють `ElectronicDeviceFactory`: наприклад, `SamsungFactory`, `AppleFactory` і т.д. Кожна з цих фабрик повинна реалізувати методи створення смартфонів та планшетів для відповідних моделей.
- Напишіть клас-демонстрацію `ElectronicDeviceDemo`, де ви створите об'єкти різних моделей смартфонів та планшетів через фабрики та продемонструєте їх функціональність.

9.5 Інтеграція різних платіжних систем

Уявіть, що ви розробляєте систему для інтеграції різних платіжних систем у вашому додатку. Ваша система має працювати з різними API платіжних систем, такими як PayPal, Stripe та інші. Вам потрібно використати паттерн Адаптер для створення адаптерів, які перетворюватимуть інтерфейси різних платіжних систем на спільний інтерфейс для вашої системи.

Вимоги

- Створіть інтерфейс `PaymentProcessor`, який містить методи, що повинні бути реалізовані всіма платіжними системами, наприклад, `processPayment()`.
- Створіть класи, що реалізують інтерфейс `PaymentProcessor` для кожної платіжної системи: наприклад, `PayPalPaymentProcessor`, `StripePaymentProcessor` і т.д. Кожен з цих класів повинен реалізувати методи, визначені в інтерфейсі `PaymentProcessor`, згідно з API відповідної платіжної системи.
- Створіть адаптери для кожної платіжної системи, які реалізують інтерфейс `PaymentProcessor`. Ці адаптери будуть містити посилання на відповідний об'єкт платіжної системи та перетворюватимуть виклики методів інтерфейсу `PaymentProcessor` на виклики методів відповідної платіжної системи.
- Напишіть клас-демонстрацію `PaymentSystemDemo`, де ви створите об'єкти адаптерів для різних платіжних систем та продемонструєте їх функціональність, викликаючи метод `processPayment()` через адаптери.

9.6 Система обробки замовлень в інтернет-магазині

Уявіть, що ви розробляєте систему обробки замовлень для інтернет-магазину. Ваша система повинна забезпечувати обробку замовлень від клієнтів, включаючи перевірку наявності товару, розрахунок вартості, створення замовлення та надсилання підтвердження клієнту. Вам потрібно використати паттерн Фасад для створення простого інтерфейсу, який буде забезпечувати ці всі функції і скривати деталі реалізації внутрішніх компонентів системи.

Вимоги:

- Створіть клас `OrderFacade`, який буде виступати як фасад для всієї системи обробки замовлень. Цей клас повинен мати методи для виконання наступних завдань:
 - Перевірка наявності товару.
 - Розрахунок вартості замовлення.
 - Створення замовлення.
 - Надсилання підтвердження клієнту.
- Клас `OrderFacade` повинен використовувати інші класи в системі, такі як `InventoryService` для перевірки наявності товарів, `PricingService` для розрахунку вартості, `OrderService` для створення замовлення та `EmailService` для надсилання електронних листів.
- Кожен з цих внутрішніх сервісів повинен мати свій власний інтерфейс та реалізацію, але клас `OrderFacade` повинен використовувати лише їхні інтерфейси, не звертаючись до конкретних класів.
- Напишіть клас-демонстрацію `OrderProcessingDemo`, де ви створите об'єкт `OrderFacade` та продемонструєте його функціональність, викликаючи різні методи фасаду.

9.7 Моделювання роботи ліфта з використанням паттерну Стан

Уявіть, що ви розробляєте програмне забезпечення для керування роботою ліфта у великому будівлі. Ваша система повинна включати в себе логіку керування рухом ліфта, управління дверима, показники поверхів та інші важливі аспекти. Для керування станами ліфта і забезпечення його правильної поведінки вам потрібно використати паттерн Стан.

Вимоги:

- Створіть клас `Elevator`, який буде представляти ліфт. Цей клас повинен мати ряд станів, таких як "Відкрито", "Закрито", "Рух вгору", "Рух вниз" та інші.
- Створіть інтерфейс `ElevatorState`, що описує методи, які повинен реалізувати кожен стан ліфта, наприклад, `openDoors()`, `closeDoors()`, `moveUp()` та `moveDown()`.
- Створіть конкретні реалізації інтерфейсу `ElevatorState` для кожного стану ліфта, де кожен з них реалізує необхідну логіку для відповідної дії.
- У класі `Elevator` створіть змінну для поточного стану ліфта та методи для переходу між станами.
- Напишіть клас-демонстрацію `ElevatorSimulation`, де ви створите об'єкт ліфта та продемонструєте його роботу, викликаючи різні методи для переміщення ліфта на різні поверхи та відкриття/закриття дверей відповідно до поточного стану.

9.8 Розв'язування квадратних та лінійних рівнянь з використанням паттерну Стратегія

Уявіть, що ви розробляєте програмне забезпечення для розв'язування квадратних та лінійних рівнянь. Ваша система повинна мати можливість розв'язувати обидва типи рівнянь за допомогою різних методів. Ви хочете, щоб ваша система була гнучкою і легко змінювалася, тому ви вирішуєте використати паттерн Стратегія.

Вимоги:

- Створіть інтерфейс `EquationSolver`, що містить метод `solveEquation()`, який буде реалізований всіма стратегіями розв'язування рівнянь.
- Створіть конкретні реалізації інтерфейсу `EquationSolver` для розв'язування квадратних та лінійних рівнянь. Наприклад, `QuadraticEquationSolver` та `LinearEquationSolver`.
- Визначте методи `solveEquation()` в кожній з реалізацій таким чином, щоб вони могли розв'язувати відповідні типи рівнянь за допомогою відповідних методів.
- Напишіть клас `EquationSolverContext`, який буде використовувати стратегію розв'язування рівнянь. Цей клас повинен мати метод `setEquationSolver()`, що дозволяє встановлювати поточну стратегію розв'язування рівнянь, а також метод `solveEquation()`, що використовує поточну стратегію для розв'язування рівняння.
- Напишіть клас-демонстрацію `EquationSolverDemo`, де ви створите об'єкт `EquationSolverContext`, встановите потрібну стратегію розв'язування та викличете метод `solveEquation()` для різних типів рівнянь.

9.9 Розглянемо інтерфейс Робітник, що моделює роботу з робітником на заводі. Робітники можуть:

- виконувати різні види робіт, за кожен з яких вони отримують певну зарплату;
- отримувати премії;
- отримувати грошові виплати від компанії для покриття витрат на побут;
- оплачувати проживання та харчування на території заводу.

Для отримання посвідчення робітника, він має набрати визначену кількість кредитів за виконані види робіт. Робітники можуть бути наступних спеціалізацій:

- зі зварювання;
- зі складування матеріалів;
- зі монтажу обладнання.

Виконання робітниками різних видів робіт здійснюється під керівництвом інженерів на заводі. При цьому інженери мають профіль відповідно до типу робіт, які вони керують: зі зварювання, зі складування матеріалів або зі монтажу обладнання. Інженери, які відповідають за зварювання, не можуть керувати робітниками зі складування матеріалів і навпаки. При цьому робітників зі складування матеріалів можуть керувати обидва типи інженерів.

Нараховування премій робітникам здійснює бухгалтерія заводу. Здійснення оплати за проживання здійснюється у відділі гуртожитку заводу, а оплату за харчування – у столовій заводу. Якщо робітник не має грошей для оплати проживання або харчування, він звільняється з заводу для того, щоб знайти засоби для існування і відповідно не отримує посвідчення робітника.

Реалізуйте шаблон програмування Відвідувач (Visitor) для моделювання процесу працевлаштування та життєдіяльності робітника до моменту отримання ним посвідчення робітника та визначіть, чи отримає він його. Вхідні дані містяться у файлах `input01.txt`, `input02.txt` і т. д. Кожен файл містить дані про одного робітника, включаючи спеціалізацію, необхідну кількість кредитів для отримання посвідчення, початковий бюджет та рядки, що описують дії робітника.

9.10 Замовлення шаурми з використанням паттерну Builder

Уявіть, що ви розробляєте систему для замовлення шаурми в онлайн-магазині. Кожен клієнт може замовити шаурму з різними інгредієнтами та додатковими складниками. Вам потрібно використати паттерн Builder для створення об'єктів замовлення шаурми з різними характеристиками.

Вимоги:

- Створіть інтерфейс `ShawarmaBuilder`, що містить методи для додавання різних інгредієнтів та налаштувань шаурми, таких як м'ясо, овочі, соуси тощо.
- Створіть конкретні реалізації інтерфейсу `ShawarmaBuilder` для різних типів шаурми, наприклад, `ClassicShawarmaBuilder`, `VeganShawarmaBuilder`, `SpicyShawarmaBuilder` тощо.
- Визначте методи для додавання різних інгредієнтів в кожному з реалізацій `ShawarmaBuilder`, які дозволяють клієнту налаштовувати замовлення згідно їхніх уподобань.
- Створіть клас `ShawarmaDirector`, який буде відповідальний за створення об'єктів замовлень шаурми з використанням відповідного білдера.
- Напишіть клас-демонстрацію `ShawarmaOrderDemo`, де ви створите об'єкти `ShawarmaDirector` та різні `ShawarmaBuilder`, встановите потрібні параметри замовлення та побудуйте об'єкти замовлень шаурми.

Список використаної літератури

1. Кренивч А. П. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" / А.П. Кренивч, О.В. Обвінцев. – К. : Видавничо-поліграфічний центр "Київський університет", 2011. – 208 с.
2. Кренивч А. П. С у задачах і прикладах : навч. посібник / А. П. Кренивч, О. В. Обвінцев. – Київ : ВПЦ "Київський університет", 2012. – 212 с.
3. Вакал Є.С.,Личман В.В.,Обвінцев О.В.,Бублик В.В.,Попов В.В. Задачі до курсу «Інформатика та програмування». [Електронний ресурс] –Режим доступу до ресурсу:
<http://matfiz.univ.kiev.ua/informatics/problems/>
4. Збірник задач з дисципліни "Інформатика і програмування" Вакал Є.С., Личман В.В., Обвінцев О.В., Бублик В.В., Довгий Б.П., Попов В.В. -2-ге видання, виправлене та доповнене –К.:ВПЦ "Київський університет", 2006.–94с
5. Кренивч А.П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни "Інформатика та програмування"–К.: ВПЦ "Київський Університет", 2017. –206с.
6. Бублик В.В. Методические указания и задания к лабораторным занятиям по курсу "Вычислительные машины и программирование" для студентов механико-математического факультета / В.В. Бублик – К. : КГУ, 1986. –60 с.