

Лабораторна робота №2.

Тема. Створення класів в Java

Мета роботи: Навчитися створювати прості класи та об'єкти на мові Java

Тривалість роботи: 4 години

Теоретичні відомості.

Клас (class) описує признаки стану та поведінку множини схожих об'єктів. Клас – це *тип даних* користувача.

Синтаксис оголошення класу:

```
<modifier>* class <class_name> {  
    <attribute_declaration>*  
    <constructor_declaration>*  
    <method_declaration>*  
}
```

Приклад класу:

```
public class Shirt {  
    public double price;  
    public void setPrice(double priceArg) {  
        price = priceArg;  
    }  
}
```

Клас - це шаблон, або проект, за яким буде зроблений об'єкт. При розробці власних класів необхідно користуватися абстрагуванням від сукупності властивостей реальних предметів, і вибирати тільки ті характеристики і властивості предмета, які задовольняють семантиці розв'язуваної задачі. Наприклад, якщо розробляється інформаційна система, яка відповідає за роздрукування Квітки при отриманні співробітниками заробітної плати, то для опису співробітника в такій системі може бути досить вказати ПІБ співробітника, розмір і дату видачі заробітної плати. Якщо ж мова йде про інформаційну систему по реєстрації та обслуговування хворих в поліклініці, то значення розміру і дати видачі заробітної плати хворих не мають ніякого значення, а ось інформація про місці їх проживання, дату народження та історії хвороби відіграють істотну роль.

Об'єкт - це мислима або реальна суть, що володіє характерні. поведінкою, відмітними характеристиками і є важливою в предметної області. Об'єктом є екземпляр класу, створений шляхом виклику конструктора класу. Кожен об'єкт має станом, поведінкою і унікальністю.

Стан (state) - сукупний результат поведінки об'єкту, одне з стабільних умов, в яких об'єкт може існувати. В будь-який конкретний момент часу стан об'єкту включає в себе перелік властивостей об'єкта і поточні значення цих властивостей.

Поведінка (behavior) - дії та реакції об'єкта, виражені в термінах передачі повідомлень і зміни стану; видима ззовні і відтворена активність об'єкта.

Унікальність (identity) - природа об'єкта; то, що відрізняє його від інших об'єктів.

Об'єкти і об'єктні змінні

Щоб працювати з об'єктами, їх потрібно спочатку створити і поставити вихідний стан. Потім до цих об'єктів застосовуються методи. У мові **Java** для створення нових екземплярів використовуються конструктори.

Конструктор - спеціальний метод, призначений для створення і ініціалізації екземпляра класу. Ім'я конструктора завжди збігається з ім'ям класу. Наприклад, конструктор класу **Employee** називається **Employee**.

Приклад, класу співробітник (**Employee**).

```
import java.util.Date;
import java.util.GregorianCalendar;

public class Employee {
    /* перерахування полів класу*/
    private String name; // ім'я
    private double salary; // розмір заробітної плати
    private Date hiredate; // дата прийняття на роботу
    public static int num_r; // номер співробітника
    // конструктор класу, завдання якого - задання значень полів класу
    {
        name = "Noname N.";
        salary = 1005.01;
        hiredate = (new GregorianCalendar()).getTime();
    }

    public Employee (String n, double s, int year, int month, int day) {
        name = n;
        salary = s;
        hiredate = (new GregorianCalendar (year, month-1, day)). getTime ();
        num_r++; //
    }

    public Employee(String in_name){
        name=in_name;
        salary=1000;
        hiredate=(new GregorianCalendar(2017,12,31)).getTime();
        num_r++;
    }

    public Employee(){
        num_r++;
    }

    // методи класу
    public String getName() { // повертає ім'я співробітника
        return name;
    }

    public double getSalary() { /* повертає розмір заробітної плати
співробітника*/
        return salary;
    }

    public Date getDate() { // повертає дату прийому на роботу
        return hiredate;
    }

    @Override
    public String toString() {
        String s;
        s = name + salary + hiredate.toString();
        return s;
    }
}
```

```

    }

    @Override
    public boolean equals(Object obj) {
        boolean b = false;
        if ( obj instanceof Employee )
        {
            Employee obj1 = (Employee) obj;
            if (    name==obj1.getName() &&
                salary==obj1.getSalary() &&
                hiredate == obj1.getDate() ) b = true;

        }
        return b;
    }
}

```

Головна програма.

```

import java.util.Scanner;
import java.io.IOException;
public class Lab3 {

    public static void main(String[] args) throws IOException {
        System.out.println("Привіт !");
        Scanner in = new Scanner(System.in);
        System.out.print("Введіть ім'я: ");
        String name = in.nextLine();
        System.out.print("Введіть розмір зарплати : ");
        int zarp = in.nextInt();
        System.out.print("Введіть рік прийому на роботу : ");
        int r = in.nextInt();
        System.out.print("Введіть місяць прийому на роботу : ");
        int m = in.nextInt();
        System.out.print("Введіть день прийому на роботу : ");
        int d = in.nextInt();
        Employee.num_r = 0;
        Employee obj = new Employee(name, zarp, r, m, d);
        System.out.println("Ваше ім'я: " + obj.getName() + " зарплата " +
obj.getSalary() );
        Employee obj1= new Employee("Степаненко С.");
        System.out.println("Ваше ім'я: " + obj1.getName() + " зарплата " +
obj1.getSalary() );
        Employee obj2= new Employee();
        System.out.println("Ваше ім'я: " + obj2.getName() + " зарплата " +
obj2.getSalary() );
        System.out.println(obj2.toString() );
        if (obj.equals(obj1)) System.out.println(" ... Так ");
        in.close();
        System.out.println(" Кількість введених працівників " + Employee.num_r
);
    }
}

```

Завдання. Розробити(створити) класи згідно варіанту мовою Java та продемонструйте їхні функції. Передбачити кілька способів ініціалізації об'єктів класу (ініціалізатори і конструктори). Клас повинен містити статичні поля і методи (продумати самостійно). В кожному класі мають бути перевантажені методи **toString** та **equal**. Проілюструвати роботу всіх методів класу.

Виконання лабораторної роботи

1. Зайти в свій обліковий запис на github.com. Зайти в github classroom.
2. Клонувати репозиторій:
ЧНУ - 241А, 241Б, 341ск - <https://classroom.github.com/a/GM828fzT>
Коледж ЧНУ - https://classroom.github.com/a/AWbtpn_G
на робочий комп'ютер.
3. Розв'язати задачі згідно варіанту.
4. В процесі написання функцій розв'язання задач лабораторної роботи періодично здійснювати синхронізацію з репозиторієм на github.com, з поясненням виконаної роботи (**git add .**, **git commit -m"коментар"**, **git push**).
5. Оформити звіт про виконання лабораторної роботи.
6. Звіт відправити в для оцінювання в <https://moodle.chnu.edu.ua> або [Google Classroom](#).

Завдання до лабораторної роботи

Завдання 1.

1. Створити клас `vector3D`, що задається трьома координатами. Обов'язково реалізувати: додавання, віднімання векторів, скалярний добуток векторів, множення на скаляр, порівняння векторів, обчислення довжини вектора, порівняння довжини векторів.
2. Створити клас `Money` для роботи з грошовими сумами. Число повинно бути представлено двома полями: типу `long` для грн. та типу `byte` – для коп. Дробова частина (копійки) при виведенні на екран повинна бути відділена від цілою комою. Реалізувати додавання, віднімання, ділення, ділення на число, операції порівняння.
3. Створити клас `Point` для роботи з точками на площині. Координати точки – декартові. Обов'язково реалізувати: переміщення вздовж осі x , вздовж осі y , визначення відстані до початку координат, знаходження відстані між двома точками, перетворення до полярних координат, порівняння на співпадання та неспівпадання.
4. Раціональний (нескоротний) дріб представлений парою цілих чисел (a, b) , де a – чисельник, b – знаменник. Створити клас `Rational` для роботи з раціональними дробами. Обов'язково реалізувати операції: додавання, віднімання, множення, ділення, порівняння (на рівність, на більше чи менше). Реалізувати доступний тільки для класу `Rational` метод скорочення (`reduce`), який викликається при виконанні арифметичних операцій.
5. Створити клас `Date` для роботи з датою у форматі «рік.місяць.день», що містить три поля типу `byte`. Клас повинен включати не менше ніж 3 конструктори: числами, рядком вигляду «рік.місяць.день» та датою. Задати

операції: обчислення дати через задану кількість днів, віднімання від дати заданої кількості днів, визначення високосності року, отримання та присвоєння окремих частин (рік, місяць, день), порівняння дат (дорівнює, до, після), обчислення кількості днів між датами.

6. Створити клас Time для роботи з часом у форматі «година:хвилина:секунда» з трьома полями типу short. Клас повинен включати не менше чотирьох конструкторів: ініціалізація числами, рядком, секундами та часом. Обов'язковими операціями є: обчислення різниці між двома моментами часу у секундах, додавання часу та заданої кількості секунд, віднімання з часу заданої кількості секунд, порівняння моментів часу, перевід в секунди, перевід в хвилини (з заокругленням до цілої хвилини).

7. Реалізувати клас Account, що представляє банківський рахунок. В класі повинні бути чотири поля: прізвище власника, номер рахунку, відсоток нарахувань та сума у грн. Відкриття нового рахунку відбувається операцією ініціалізації. Необхідно виконати такі операції: зміна власника, зняти деяку суму грошей з рахунку, покласти гроші на рахунок, начислити відсотки, отримати суму прописом.

8. Створити клас Goods (товар) з полями: назва товару, дата оформлення, ціна товару, кількість одиниць товару, номер накладної, по якій товар поступив на склад. Реалізувати методи зміни ціни товару, зміни кількості, обчислення вартості товару.

9. Створити клас Payment (зарплата), в якому є поля: прізвище, ім'я, оклад, рік поступлення на роботу, відсоток надбавки, податок, кількість відпрацьованих днів в місяць, кількість днів в місяці, нарахована та утримана сума. Реалізувати методи: обчислення нарахованої суми, обчислення утриманої суми, обчислення стажу роботи (кількості років від початку роботи до поточного року).

10. Реалізувати клас Cursor. Полями є координати курсору по горизонталі та вертикалі – цілі додатні числа, вигляд курсору – горизонтальний чи вертикальний, розмір курсору – ціле значення від 1 до 15. Реалізувати методи зміни координат курсору, зміни вигляду курсору, зміни розміру курсору, метод приховання та відновлення курсору.

11. Створити клас Fraction для роботи з дробовими числами. Число повинно бути представлено двома полями: ціла частина – довге число, дробова частина – коротке ціле. Реалізувати арифметичні операції додавання, віднімання, множення та операції порівняння.

12. Створити клас Money для роботи з грошовими одиницями. Номінали гривень можуть набувати значень 1, 2, 5, 10, 20, 50, 100, 200, 500. Копійки представити як 0.01 (1 копійка), 0.02 (2 копійки), 0.05 (5 копійок), 0.1 (10

копійок), 0.25 (25 копійок), 0.5 (50 копійок). Сума повинна бути представлена полями-номіналами, значеннями яких повинна бути кількість купюр даного номіналу. Реалізувати додавання сум, віднімання сум, ділення сум, ділення суми на дробове число та операції порівняння. Копійки при виводі на екран повинні відділятися від грн. комою.

13. Створити клас `Bankomat`. В класі повинні міститися поля для зберігання ідентифікаційного номера банкомату, інформації про поточну суму грошей, що є в банкоматі, мінімальної та максимальної сум, які можна зняти клієнту за один день. Реалізувати метод ініціалізації банкомата, метод завантаження купюр в банкомат та метод зняття певної суми грошей. Метод зняття повинне виконувати перевірку на коректність суми, що знімається.

14. Реалізувати клас `Triangle` для представлення трикутника. Поля повинні включати кути та сторони. Реалізувати операції: отримання та зміни полів даних, обчислення площі, обчислення периметра, отримання висот, а також визначення вигляду трикутника (рівносторонній, рівнобедрений або прямокутний).

15. Створити клас `Angle` для роботи з кутами на площині, які задаються величиною в градусах та хвилинах. Обов'язково повинні бути реалізовані: перевід в радіани, приведення до діапазону 0.00-360.00, збільшення та зменшення кута на задану величину, отримання синусу, порівняння кутів.

Завдання 2.

1. Клас «комплексне число», який дозволяє знаходити модуль комплексного числа, його аргумент, виконує основні арифметичні операції з числами: додавання, віднімання, множення, ділення, комплексного спряження.

2. Клас «вектор», який описує вектор у 3-вимірному декартовому просторі:
- вектор задається трьома координатами;
- методи класу дозволяють знаходити модуль вектора, множити вектор на число,
- додавати вектори, множити скалярно і векторно на інший вектор.

3. Клас «матриця 2x2», який описує матрицю з 2 стовпчиками і 2 рядками. Методи класу дозволяють знаходити детермінант, обернену матрицю, множити на іншу матрицю, додавати до іншої матриці, множити матрицю на число.

4. Клас «трикутник»:

- трикутник задається довжиною трьох сторін
- конструктор класу перевіряє, чи можна створити трикутник із заданими сторонами.
- методи класу дозволяють знаходити периметр трикутника, площу трикутника;

- існує метод для порівняння з іншим трикутником.

5. Клас «квадрат», який описує квадрат з заданими сторонами. Методи класу дозволяють знаходити периметр та площу квадрата, довжину діагоналей.

6. Клас «коло», який описує коло на координатній площині:

- коло задається радіусом та координатами центра;
- методи класу дозволяють знаходити довжину кола, площу круга;
- існує метод, який визначає, чи належить точка з координатами (x,y) колу;
- існує метод, який визначає кількість точок перетину з іншим колом.

7. Клас «куб», який описує куб з заданим ребром. Методи класу дозволяють знаходити сумарну площу граней, об'єм куба, довжину великої діагоналі.

8. Клас «конус», який описує правильний конус з заданою висотою та радіусом основи. Методи класу дозволяють знаходити площу бічної основи та об'єм конуса.

9. Клас «призма», який описує прямокутну призму заданої висоти, в основі якої лежить правильний трикутник з заданою стороною. Методи класу дозволяють знаходити площу бічної основи та об'єм призми.

10. Клас «прямокутник»:

- прямокутник задається довжиною двох сторін;
- методи класу дозволяють знаходити периметр трикутника, площу прямокутника;
- існує метод для порівняння з іншим прямокутником;
- існує метод для виявлення подібності до іншого прямокутника.

11. Клас «квадратне рівняння»:

- квадратне рівняння задається трьома числами (a,b,c), які описують рівняння $ax^2 + bx + c = 0$;
- існує метод для обчислення кількості коренів рівняння;
- існує метод для обчислення коренів, який повертає масив з одного або двох чисел або NULL.

12. Клас «паралелограм»:

- паралелограм задається довжинами двох сторін та кутом між сторонами;
- існують методи для обчислення периметра та площі паралелограма;
- існує метод для порівняння з іншим паралелограмом;
- існує метод для визначення подібності до іншого паралелограма.

13. Клас «пряма на координатній площині»:

- пряма задається трьома числами (a,b,c) у рівнянні $ax + by + c = 0$;
- існує метод для визначення, чи належить точка (x,y) прямій;

- існує метод для визначення координат (x,y) точки перетину з іншою прямою.

Завдання 3.

1. Створити клас Drib, у якому реалізувати додавання, віднімання, множення, ділення і скорочення простих дробів. Результат завжди повинен бути нескоротним. На вхід програма отримує 4 цілих числа (чисельник і знаменник першого і другого дробів).
2. Створити клас VeryLongNumber, в якому реалізувати додавання, віднімання, множення, ділення і піднесення в натуральний степінь дуже довгих чисел. На вхід програма отримує два рядки (у командному рядку). В кожному рядку містяться 2 довгих цілих числа.
3. Створити клас календар, в якому реалізувати пошук дня місяця і тижня для введеного числа і року (наприклад для вхідних даних 32, 2002 повернути 1 лютого, п'ятниця)
4. Створити клас Math, у якому реалізувати пошук найбільшого спільного дільника та найменшого кратного двох натуральних чисел. На вхід програма отримує 2 цілих числа у командному рядку.
5. Створити клас Baloon, в якому реалізувати пошук об'єму перетину та об'єднання 2 куль. Кожна куля задається 3 цілими числами (координатами центру і радіусом). На вхід програма у командному рядку отримує 6 чисел, які задають дві кулі.
6. Створити 2 масиви цілих випадкових чисел, упорядкувати по зростанню їх методом бульбашки і злити в третій, так щоб він був впорядкованим по зростанню.
7. Створити масив цілих випадкових чисел. Впорядкувати їх за допомогою методу швидкого сортування.
8. На вхід задається арифметичний вираз, який містить цілі невід'ємні числа і операції +, -, *. Обчислити та надрукувати значення цього виразу враховуючи пріоритет операцій.
9. Створити клас Complex, в якому реалізувати додавання, віднімання, множення, ділення, пошук модуля, аргументу. На вхід програма отримує в командному рядку 4 числа (дійсну і уявну частини першого і другого чисел).
10. Організувати динамічну структуру даних чергу з цілих чисел. Реалізувати додавання елементів у чергу, вилучення їх, пошук потрібного елемента.
11. Організувати динамічну структуру даних стек з цілих чисел. Реалізувати додавання елементів у стек, вилучення їх, пошук потрібного елемента.
12. Організувати динамічну структуру даних бінарне дерево з цілих чисел. Реалізувати додавання елементів у дерево, вилучення їх, пошук потрібного елемента.