

ПРАКТИЧНА РОБОТА 3

Поліморфізм, інтерфейси та узагальнене програмування

Система оцінювання

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Завдання на практичну роботу	3,6*
3.	<i>Здача звіту</i>	0,4
	Всього	5

* – діє бонусна система

1. *0,3 бала* Написати клас, у якому буде присутній узагальнений метод printArray для друку елементів масивів. Метод main() повинен містити три масиви для тестування:

```
Integer[] intArray = {1, 2, 3, 4, 5};
Double[] doubleArray = {1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.7};
Character[] charArray = {'H', 'E', 'L', 'L', 'O'};
```

Виконайте перевантаження (overloading) узагальненого методу printArray так, щоб неузагальнена версія методу друкувала конкретно масиви елементів типу String у наступному форматі

```
Array stringArray contains:
one      two      three    four
five     six      seven    eight
```

2. *0,4 бала* Реалізувати інтерфейс Меблі, абстрактний клас Шафа та клас Книжкова шафа, а також наслідування та поліморфізм для них. Опишіть в інтерфейсі один або більше методів за замовчуванням.
3. *0,4 бала* Створіть інтерфейс Measurable з методом double getMeasure(), який вимірює об'єкт деяким чином. Клас Employee має реалізовувати Measurable. Забезпечте метод double average(Measurable[] objects), який обчислює усереднену величину вимірів. Використайте метод для обчислення усередненої заробітної плати з масиву робітників. Доповніть інтерфейс Measurable методом largest(Measurable[] objects). Використайте його для знаходження імені робітника з найвищою заробітною платою.
4. *0,4 бала* Напишіть узагальнений метод selectionSort(), який виконуватиме сортування вибором Integer та Float масивів. У відповідному класі вводьте дані в масив з клавіатури, запускайте сортування та виводьте його результат.
 - Підказка: використовуйте <T extends Comparable<T>> для методу selectionSort(), щоб можна було використовувати метод compareTo() для порівняння об'єктів, представлених типом T.
 - Псевдокод алгоритму

```

// sort array using selection sort
public static void selectionSort(int[] data)
{
    // loop over data.length - 1 elements
    for (int i = 0; i < data.length - 1; i++)
    {
        int smallest = i; // first index of remaining array

        // loop to find index of smallest element
        for (int index = i + 1; index < data.length; index++)
            if (data[index] < data[smallest])
                smallest = index;

        swap(data, i, smallest); // swap smallest element into position
        printPass(i + 1, smallest); // output pass of algorithm
    }
} // end method selectionSort

```

5. 0,5 бала Напишіть класи Shape, Circle, Rectangle, Square відповідно до діаграми класів.



Клас Shape має бути абстрактним та містити:

- Дві protected instance variables: color(String) та filled(boolean), що позначені на діаграмі символом #.
- Геттери та сеттери для всіх instance variables, а також метод toString().
- Два абстрактні методи getArea() та getPerimeter(), виділені курсивом на діаграмі.

Підкласи Circle та Rectangle мають переозначати абстрактні методи getArea() та getPerimeter(), а також забезпечувати правильну реалізацію. Також у них переозначається метод toString().

Додайте тестовий клас, що міститиме наступний код:

```
Shape s1 = new Circle(5.5, "RED", false); // Upcast Circle to Shape
System.out.println(s1);                  // which version?
System.out.println(s1.getArea());         // which version?
System.out.println(s1.getPerimeter());    // which version?
System.out.println(s1.getColor());
System.out.println(s1.isFilled());
System.out.println(s1.getRadius());

Circle c1 = (Circle)s1;                  // Downcast back to Circle
System.out.println(c1);
System.out.println(c1.getArea());
System.out.println(c1.getPerimeter());
System.out.println(c1.getColor());
System.out.println(c1.isFilled());
System.out.println(c1.getRadius());

Shape s2 = new Shape();

Shape s3 = new Rectangle(1.0, 2.0, "RED", false); // Upcast
System.out.println(s3);
System.out.println(s3.getArea());
System.out.println(s3.getPerimeter());
System.out.println(s3.getColor());
System.out.println(s3.getLength());

Rectangle r1 = (Rectangle)s3; // downcast
System.out.println(r1);
System.out.println(r1.getArea());
System.out.println(r1.getColor());
System.out.println(r1.getLength());

Shape s4 = new Square(6.6); // Upcast
System.out.println(s4);
System.out.println(s4.getArea());
System.out.println(s4.getColor());
System.out.println(s4.getSide());

// Take note that we downcast Shape s4 to Rectangle,
// which is a superclass of Square, instead of Square
Rectangle r2 = (Rectangle)s4;
```

```

System.out.println(r2);
System.out.println(r2.getArea());
System.out.println(r2.getColor());
System.out.println(r2.getSide());
System.out.println(r2.getLength());

```

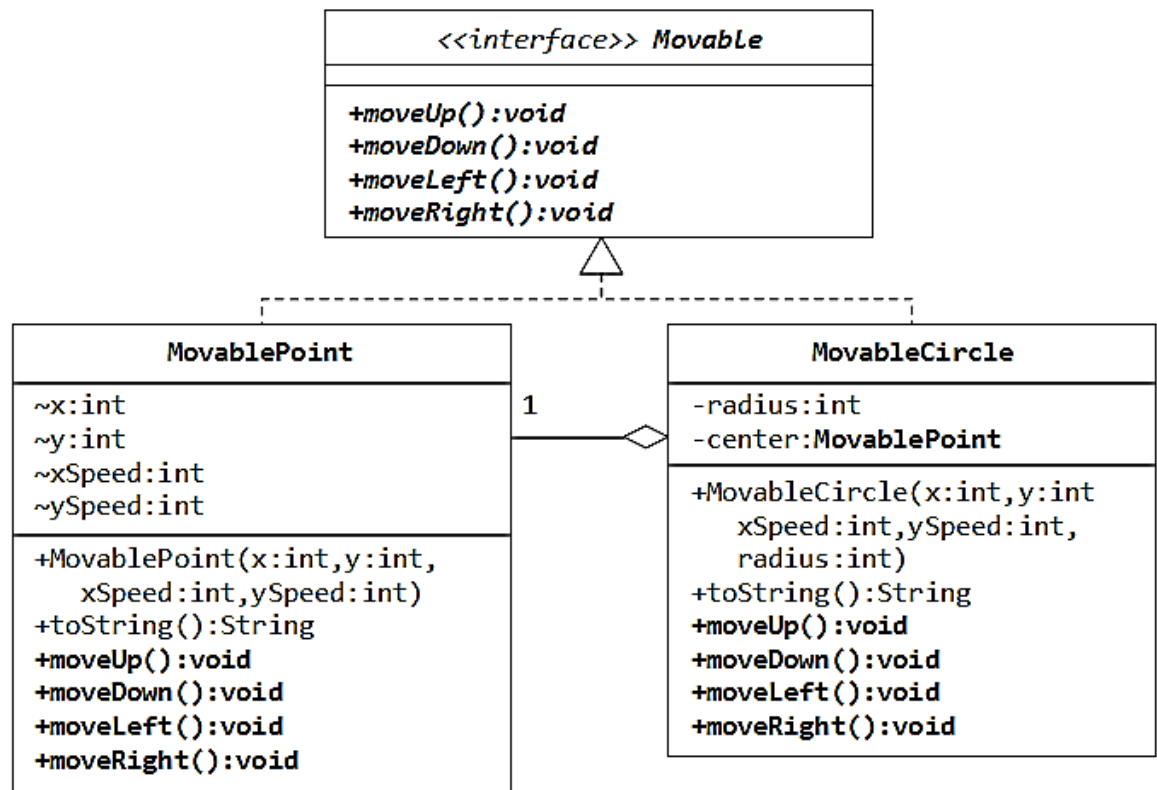
```

// Downcast Rectangle r2 to Square
Square sq1 = (Square)r2;
System.out.println(sq1);
System.out.println(sq1.getArea());
System.out.println(sq1.getColor());
System.out.println(sq1.getSide());
System.out.println(sq1.getLength());

```

Деякі інструкції можуть викликати помилки компіляції. Опишіть причини їх виникнення та прокоментуйте вивід.

6. *0,5 бала* Припустимо, що Ви маєте набір об'єктів з подібною поведінкою: вони можуть рухатись вгору, вниз, вправо або вліво. Конкретна поведінка (як рухатись та на яку відстань) залежить від самих об'єктів. Поширеним способом моделювання такої поведінки є визначення інтерфейсу Movable з абстрактними методами moveUp(), moveDown(), moveLeft() та moveRight(). Класи, які реалізують інтерфейс Movable, будуть постачати конкретні реалізації абстрактних методів.



Напишіть два конкретних класи MovablePoint та MovableCircle, які реалізують інтерфейс Movable відповідно до діаграми класів.

Реалізуйте тестовий клас та спробуйте додати в нього наступні інструкції:

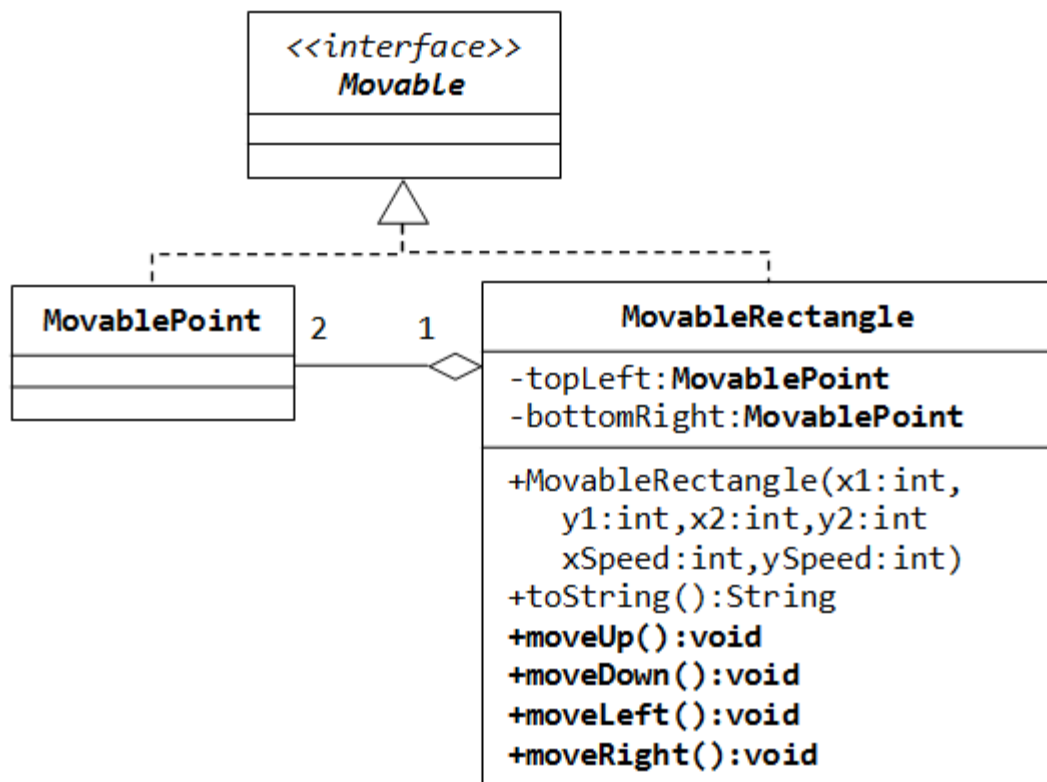
```

Movable m1 = new MovablePoint(5, 6, 10);    // upcast
System.out.println(m1);
m1.moveLeft();
System.out.println(m1);

Movable m2 = new MovableCircle(2, 1, 2, 20); // upcast
System.out.println(m2);
m2.moveRight();
System.out.println(m2);

```

Додайте новий клас `MovableRectangle`, який компонує два `MovablePoints` (верхній лівий та нижній правий кути) та реалізує інтерфейс `Movable`, як це показано на діаграмі. Переконайтесь, що обидві точки мають однакову швидкість.



7. *0,4 бала* (Лямбда-вирази) Створіть фіктивний функціональний інтерфейс, який представлятиме слухача події (event listener). Дана подія повинна бути типу `Object` та передаватись у слухача. Запишіть реалізацію даного інтерфейсу у стилі Java 7 (анонімний інтерфейс) та Java 8. Реакція на подію має виводитись на екран.
8. *0,4 бала* (Предикати) Функціональний інтерфейс `Predicate` визначено в пакеті `java.util.Function`. Він покращує керуваність коду, допомагає в модульному тестуванні коду та містить кілька методів, зокрема:
 - `isEqual(Object targetRef)`: повертає предикат, який тестує рівність двох аргументів за допомогою методу `Objects.equals(Object, Object)`.
 - `and(Predicate other)`: повертає складений предикат, який представляє логічний оператор AND для двох предикатів
 - `negate()`: логічне заперечення предикату.

- `or(Predicate other)`: повертає складений предикат, який представляє логічний оператор OR для двох предикатів.
- `test(T t)`: визначає значення предикату, враховуючи переданий аргумент `argument.boolean test(T t)`.

Напишіть клас `Employee`, який працюватиме з ІД робітника, його віком, стажом роботи, заробітною платою, статтю та ПІБ. Створіть масив з 10 робітників та надайте їм довільну інформацію, яка відповідатиме полям класу. Доповніть клас методами-предикатами, які перевірятимуть введених робітників за наступними критеріями:

- вихід працівника на пенсію: вік чоловіків понад 65 років, стаж роботи – понад 35 років; для жінок 60 та 30 років відповідно;
- нарахування доплати за вислугу років (понад 15 років стажу дає +15% до заробітної плати);

Виведіть інформацію по кожному працівнику з масиву на екран разом з результатами роботи предикатів.

9. *0,3 бала* (Функціональні інтерфейси) Напишіть узагальнений функціональний інтерфейс, єдиний абстрактний метод якого займається реверсивним виводом переданого йому значення. Створіть реалізації для реверсивного виводу рядка та цілого числа, виведіть відповідні значення для довільно обраного тексту та цілого числа.