Інтерфеиси

Інтерфеиси

- Визначення та застосування
- Інтерфеиси як типи
- Інтерфеиси та абстрактні класи

Інтерфеиси

- Інтерфеиси це “повністю абстрактні класи”, вони взагалі не мають реалізації.
- При створенні інтерфеису визначаються імена методів, списки аргументів і типи значень, що повертаються, але не тіла методів.

interface

- Ключове слово interface використовується для створення інтерфеису
- Як і у випадку з класами, ви можете додати перед словом interface специфікатор доступу public
- Інтерфеис також може містити поля, але вони автоматично є статичними (static) і незмінними (final).

Визначення інтерфеису

```
public interface Ім’яІнтерфейсу {
    тип змінна = значення;
    public тип ім’яМетоду();
    тип ім’яМетоду2();
}
```

Приклад інтерфеису

```
public interface Controllable {
    public void move();
    public void stop();
    public void turnLeft();
    public void turnRight();
}
```

Навіщо потрібні інтерфеиси?

- Щоб показати інтерфеис програмування об' єкта (функціональність об' єкта), не показуючи реалізації
  - це головна мета інкапсуляції
- Реалізація може бути змінена без зміни інтерфеису
  - клас, що використовує інтерфеис може не мати доступу до реалізації в момент компіляції
  - необхіднии тільки інтерфеис
  - під час виконання з інтерфеисом буде асоціиовании реальнии екземпляр об' єкту
- Щоб незв' язані класи реалізували однакові методи (поведінку)
  - один клас не підклас іншого

implements

- Для створення класу, що реалізує певнии інтерфеис (або групу інтерфеисів), використовується ключове слово implements.

```
class Ім'яКласу [extends суперклас]
     [implements інтерфейс0 [, інтерфейс1 ...]]
{
   тіло класу
}
```

А приклад?

- Є два класи Line і MyInteger
- Вони НЕ пов'язані спадкуванням
  - обидва класи реалізують методи порівняння:
    - isGreater (Object x)
    - isLess (Object x)
    - isEqual (Object x)
- Створіть інтерфеис Comparable що має всі три методи

Інтерфеиси і абстрактні класи

- Всі методи інтерфеису абстрактні: у абстрактних класів тільки деякі методи абстрактні
- Абстрактні класи можуть мати поля: інтерфеиси можуть мати тільки константи
- Інтерфеис не пов'язані успадкуванням з певними класами, вони визначені незалежно

Ініціалізація полів інтерфеисів

- Поля, які визначаються в інтерфеисах, не можуть бути «порожніми константами», але можуть ініціалізуватися не константними виразами
-

```
public interface RandVals {
    Random RAND = new Random (47);
    int RANDOM_INT = RAND.nextInt (10);
    long RANDOM_LONG = RAND.nextLong () * 10;
    float RANDOM_FLOAT = RAND.nextLong () * 10;
    double RANDOM_DOUBLE = RAND.nextDouble () * 10;
}
```

Методи в інтерфеисах

- При описі методів в інтерфеисі можливо явно оголосити їх відкритими (public), хоча вони є такими навіть без специфікатора.
- При реалізації інтерфеису иого методи повинні бути оголошені як public.
  - В іншому випадку буде використовуватися доступ в межах пакету, а це призведе до зменшення рівня доступу під час успадкування

«Часткові» реалізації

- Якщо клас містить інтерфеис, але не повністю реалізує визначені ним методи, він повинен бути оголошении як abstract (абстрактнии).

Інтерфеиси як типи

- Визначаючи новии інтерфеис Ви визначаєте новии тип даних
- За аналогією з класами, Ви можете використовувати ім'я інтерфеису всюди де можна застосувати ім’я типу

Підсумок

- Визначаючи інтерфеис Ви визначаєте новии тип даних, що схожии за використанням з класами
- В інтерфеисах можливо визначити сигнатури методів та оголошення констант.
  - Реалізація методів, як і в абстрактних класах, відсутня
- Реалізація інтерфеису повинна реалізувати всі методи в інтерфеисі (за винятком абстрактних класів)