

## Przestrzeń krotek

### Linda — ogólna koncepcja

- Mechanizm komunikacji międzyprocesowej zaproponowany przez Davida Gelerntera w 1985
- Luźne powiązanie komunikujących się procesów (nie muszą znać się wzajemnie, nie muszą działać jednocześnie — komunikacja nieustanna)
- Asocjacyjna identyfikacja komunikatów (w odróżnieniu do kolejkowania) we współdzielonej przestrzeni
- Współczesne implementacje:
  - ↳ JavaSpaces — Sun Microsystems (w ramach technologii Jini, projekt przejęty przez ASF — Apache River), komercyjna implementacja dostarczana przez Gigaspaces
  - ↳ TSpaces — IBM

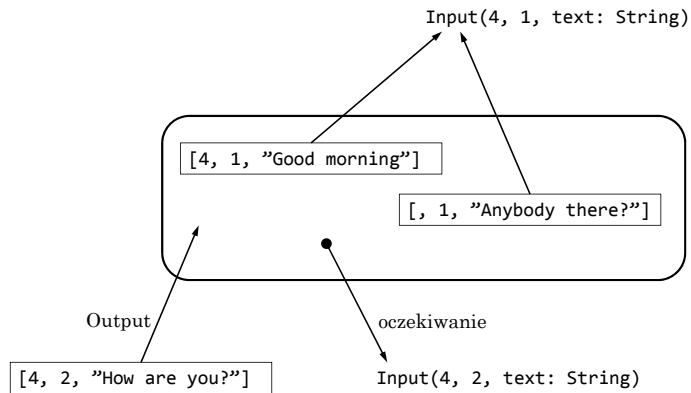
### Linda — podstawowe pojęcia

- Krotka — uporządkowana kolekcja danych określonych typów — atrybutów, przy czym atrybuty mogą (ale nie muszą) mieć nadaną konkretną wartość
- Przestrzeń krotek — wspólne miejsce dostępne kooperujących procesów, gdzie gromadzone są krotki
- Interfejs dostępu do przestrzeni krotek
  - ↳ Output — umieszczanie krotki w przestrzeni
  - ↳ Input — pobieranie krotki z przestrzeni
  - ↳ Read — odczytywanie krotki bez pobierania (odczytana krotka w dalszym ciągu pozostaje w przestrzeni)
  - ↳ Try\_Input, Try\_Read — nieblokujące wersje Input i Read

### Linda — realizacja operacji dostępu

- Krotka staje się dostępna w przestrzeni po wykonaniu operacji Output, której parametrami są wartości atrybutów, np.:  
Output(4, 1, "Good morning")
- Operacja Input powoduje pobranie (usunięcie z przestrzeni) krotki, której wartości atrybutów są zgodne z parametrami operacji, np.:  
Input(4, 1, text: String)  
Wartości pozostałych atrybutów zostaną nadane zgodnie z zawartością pobranej krotki.
- Jeśli krotka została umieszczona w przestrzeni bez podania wartości któregoś z atrybutów (np.  
Output(, 1, "Anybody there?")  
może ona zostać pobrana przez Input z dowolną wartością tego atrybutu lub pominięciem tej wartości.

## Linda — operacje na przestrzeni krotek



## Linda — specyfikacja komunikacji

- Przestrzeń krotek jest zbiorem (nie kolejka) — krotki nie są uporządkowane i mogą być odbierane w innej kolejności niż były umieszczane.
- W realizacji operacji dostępu gwarantowana jest ogólnie rozumiana żywość:
  - ↳ Jeśli procesy czekają (w operacji Input) na krotkę, to przy odpowiednio dużej liczbie umieszczonych krotek z oczekiwanymi wartościami atrybutów każdy w końcu ją otrzyma.
  - ↳ Jeśli krotka jest w przestrzeni, to po odpowiednio dużej liczbie operacji Input (ze zgodnymi parametrami) zostanie w końcu odczytana.

## JavaSpaces (przestrzeń obiektów)

- Odpowiednikiem krotki jest obiekt klasy implementującej interfejs Entry.
- Zmienne instancje obiektu-krotki muszą być publiczne.
- Obiekt-krotka ma ustalony czas życia w przestrzeni (czas wynajmowania przestrzeni — lease time).
- Czas życia można zwiększać nawet po umieszczeniu obiektu-krotki w przestrzeni.
- Operacje na przestrzeni krotek realizowane są poprzez metody write, take, read, takeIfExists, readIfExists interfejsu JavaSpace.

## Przykład definicji klasy dla obiektu-krotki

```
public class Message implements Entry {
    public Integer id;
    public Integer num;
    public String text;

    public Message () {}
    public Message (Integer i, Integer n, String t){
        id = i;
        num = n;
        text = t;
    }
}
```

### Przykład umieszczania krotki w przestrzeni

```
JavaSpace space = getSpace();  
msg = new Message ( new Integer(4),  
                    new Integer(counter++),  
                    "Good morning");  
Lease l = space.write(msg, null, 6*60*60*1000);
```

### Przykład pobierania krotki z przestrzeni

```
JavaSpace space = getSpace();  
template = new Message ( new Integer(4),  
                          null,  
                          null);  
Message msg = (Message)space.read( template,  
                                   null,  
                                   60*60*1000);
```