Programowanie współbieżne i rozproszone

WYKŁAD 7

Komunikacja między obiektami

Najbardziej podstawowa komunikacja między obiektami w języku Java polega na wywoływaniu metod.

Historia

- Protokół zdalnego wywoływania procedur, stworzony został przez firmę Sun i swego czasu był dość popularny w systemach z rodziny Unix.
- Już język C umożliwiał zdalne wywoływanie procedur RPC (*Remote Procedure Call*).
- Mechanizm RMI został wprowadzony w języku Java w wersji 1.1.

Zdalne wywoływanie metod

- RMI (*Remote Method Invocation*) – zdalne wywoływanie metod.
- RMI wspiera programowanie rozproszone dzięki możliwości wywoływania (wykonywania) metod obiektów znajdujących się na odległych komputerach.

Zdalne wywoływanie metod

- RMI umożliwia również łatwe tworzenie oprogramowania w modelu klient-serwer bez konieczności wchodzenia na poziom programowania sieciowego.
- W programowaniu sieciowym programista musi zapewnić właściwą komunikację pomiędzy częściami programu wykonującymi się na różnych komputerach przy wykorzystaniu tradycyjnych metod komunikacji (→ gniazda).

Zdalne wywoływanie metod

- Dzięki RMI możliwe jest wykonywanie fragmentów programu w obrębie różnych maszyn wirtualnych, działających na różnych komputerach (połączonych siecią), które mogą pracować pod kontrolą różnych systemów operacyjnych.

Zdalne wywoływanie metod

- Wywoływanie metod obiektów znajdujących się na odległych komputerach nie różni się prawie niczym od wywoływania metod obiektów lokalnych.
- Wymagane są jedynie czynności przygotowawcze wykonywane jednokrotnie.

Zdalne wywoływanie metod

- Przy wykorzystaniu RMI mogą być tworzone zarówno programy w modelu klient-serwer jak i programy rozproszone bez wyróżnionej części centralnej (serwera).
- W drugim przypadku realizowana jest komunikacja pomiędzy rozproszonymi obiektami składającymi się na program.

Zdalne wywoływanie metod

- Parametry aktualne wywołań metod zdalnych obiektów są serializowane (klasy tych obiektów muszą implementować interfejs **`Serializable`**) i przekazywane do serwera wraz z zadaniem wywołania metody.
- Podobnie serializowane są obiekty – wyniki zwracane do miejsca wywołania.

Elementy RMI

- **Obiekt zdalny**
  - obiekt, którego klasa implementuje interfejs **`Remote`**
  - obiekt udostępniający usługi realizowane przez metody

Elementy RMI

- **Rejestr**
  - kolekcja przechowująca referencje do zdalnych obiektów i ich nazwy, pod którymi zostały zarejestrowane
  - dostęp do obiektu zdalnego możliwy jest poprzez zarejestrowaną jego nazwę

Elementy RMI

- **Rejestr**
  - rejestr jest zdalnym obiektem tworzonym za pomocą metody

```
static Registry createRegistry(int port)
```

klasy

```
LocalRegistry
```

Elementy RMI

- **Rejestr**
  - wybrane metody:
    - **`void bind(String nazwa, Remote obiekt)`** – rejestracja obiektu pod podaną nazwą (jeśli podana nazwa jest w użyciu zgłaszany jest wyjątek **`AlreadyBoundException`**)
    - **`void rebind(String nazwa, Remote obiekt)`** – rejestracja obiektu pod podaną nazwą (jeśli podana nazwa jest w użyciu nowy obiekt zastępuje stary)

Elementy RMI

- **Rejestr**
  - wybrane metody (cd.):
    - **`void unbind(String nazwa)`** – wyrejestrowuje obiekt o podanej nazwie (gdy nazwa nie istnieje zgłaszany jest wyjątek Not**`BoundException`**)
    - **`Remote lookup(String nazwa)`** – zwraca referencję do obiektu o podanej nazwie

Elementy RMI

- **Rejestr**
  - format łańcucha nazw rejestrowanych obiektów:

```
rmi://host:port/klasa_obiektu
```

Elementy RMI

- **Interfejs**
  - deklarowanym typem odniesienia do zdalnego obiektu jest interfejs implementowany przez klasę tego obiektu
  - interfejs implementowany przez klasę zdalnego obiektu musi dziedziczyć z interfejsu *Remote* i musi deklarować metody, które będą zdalnie wywoływane

Elementy RMI

- **Namiastka**
  - obiekt pośredniczący w wywołaniach metod zdalnego obiektu
  - klasa namiastki przesyłana jest do programu, który wywołuje metodę obiektu zdalnego
  - obiekt namiastki tworzony jest automatycznie
  - obiekt namiastki reprezentuje lokalnie zdalny obiekt

Elementy RMI

- **Zarządca bezpieczeństwa**
  - nadzoruje ładowanie klas zdalnych obiektów