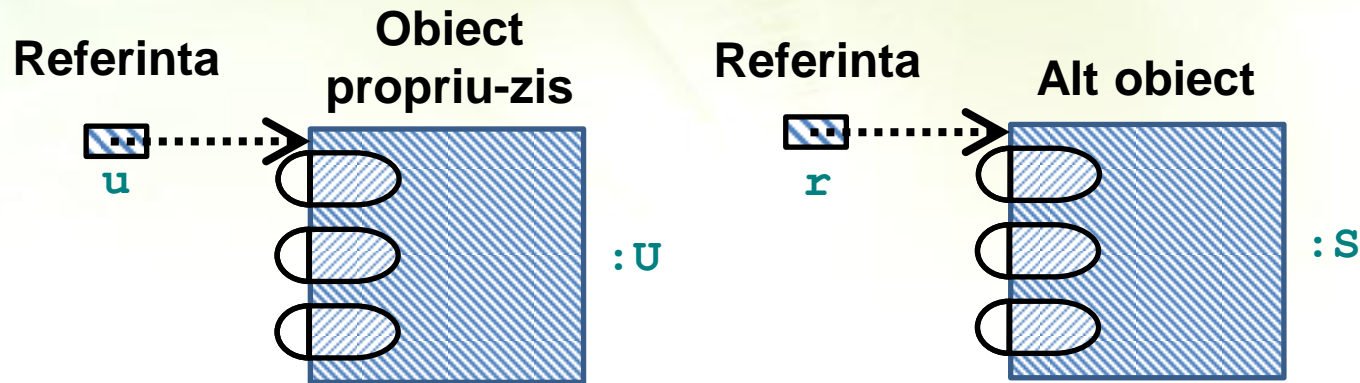


Obiecte si clase

- Crearea si legarea obiectelor

Crearea obiectelor

In cate moduri poate un obiect sa obtina o referinta catre un alt obiect?

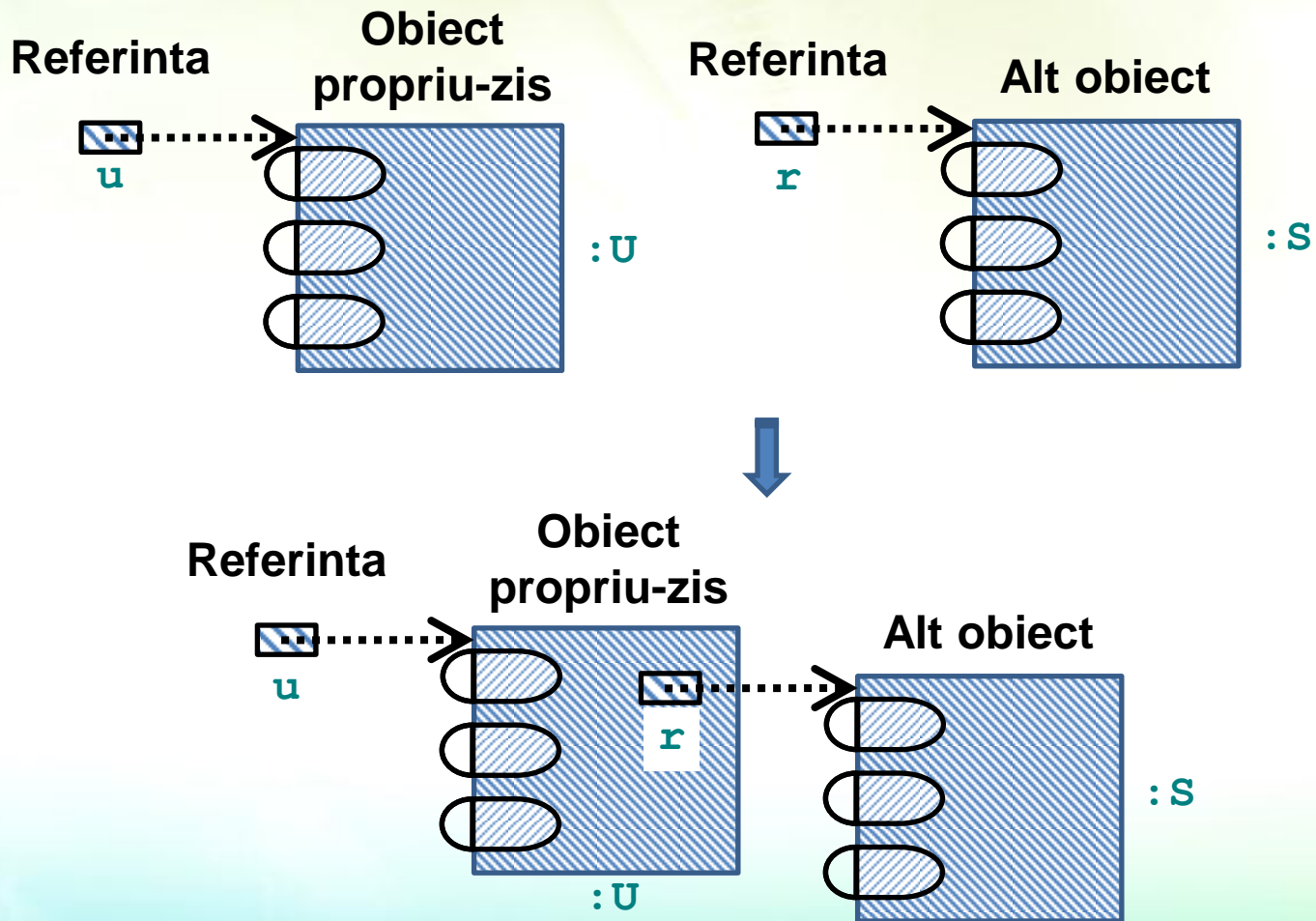


Intre un obiect **utilizator** (din clasa **U**) si un obiect **tinta** (din clasa **S**)

- se realizeaza o **legatura dinamica**
 - printr-o **referinta (r)** catre obiectul **tinta detinuta de** obiectul **utilizator**
 - si **apelul** unei **metode** a obiectului **tinta (a.k.a. trimitere de mesaj)**

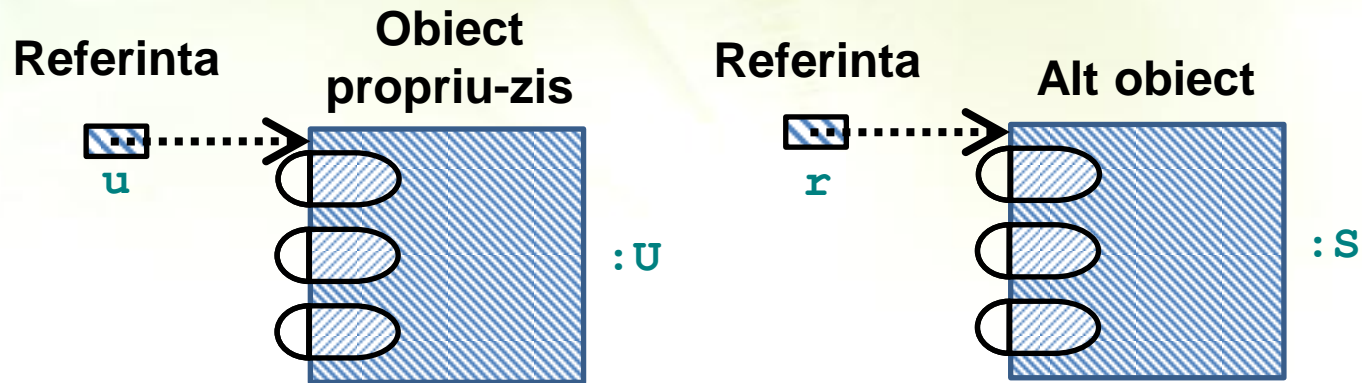
Crearea obiectelor

In cate moduri poate un obiect sa obtina o referinta catre un alt obiect?



Crearea obiectelor

In cate moduri poate un obiect sa obtina o referinta catre un alt obiect?



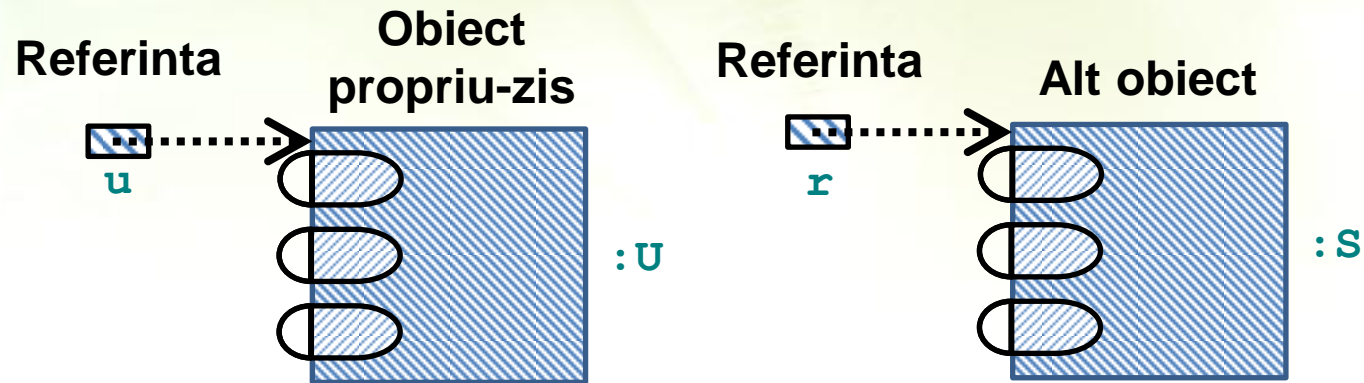
Codurile necesare in orice situatie

```
public class U { // clasa primului obiect
    private S r; // referinta catre al doilea obiect
}

public class S { // clasa celui de-al doilea obiect
}
```

Crearea obiectelor

In cate moduri poate un obiect sa obtina o referinta catre un alt obiect?



Legarea obiectelor poate fi facuta

- de catre **primul obiect singur**
 - **daca il creeaza pe al doilea (1)**
- de catre **primul obiect ajutat de un obiect tert**
 - **daca il primeste de la tert - in momentul crearii (2)**
 - **pe parcurs (3)**
- **daca il cere de la tert pe parcurs (4)**

Crearea obiectelor

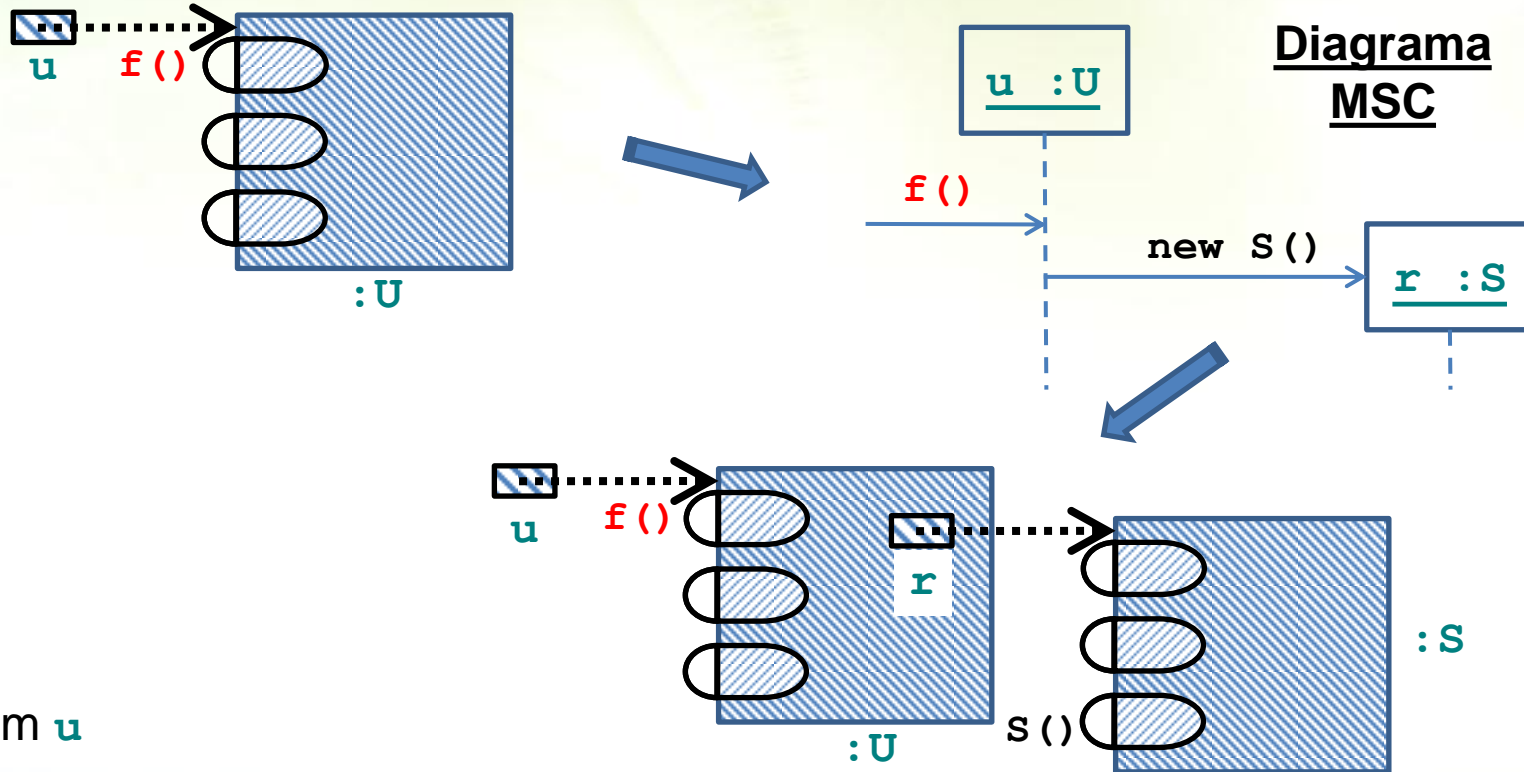
1. Primul obiect il creeaza pe al doilea

```
public class U { // clasa primului obiect
    private S r;
    ... f() {
        ...
        r = new S(); // crearea celui de-al doilea obiect
    }
}

public class S { // clasa celui de-al doilea obiect
    // constructor S() explicit sau implicit
}
```

Crearea obiectelor

1. Primul obiect il **creeza** pe al doilea



Acum `u`

- il poate **utiliza** pe `r`
- trimittandu-i **mesaje** (apelandu-i **metode**)
- prin care ii **deleaga responsabilitati**

Crearea obiectelor

2. Primul obiect il **primeste** pe al doilea **in momentul crearii** de la un tert

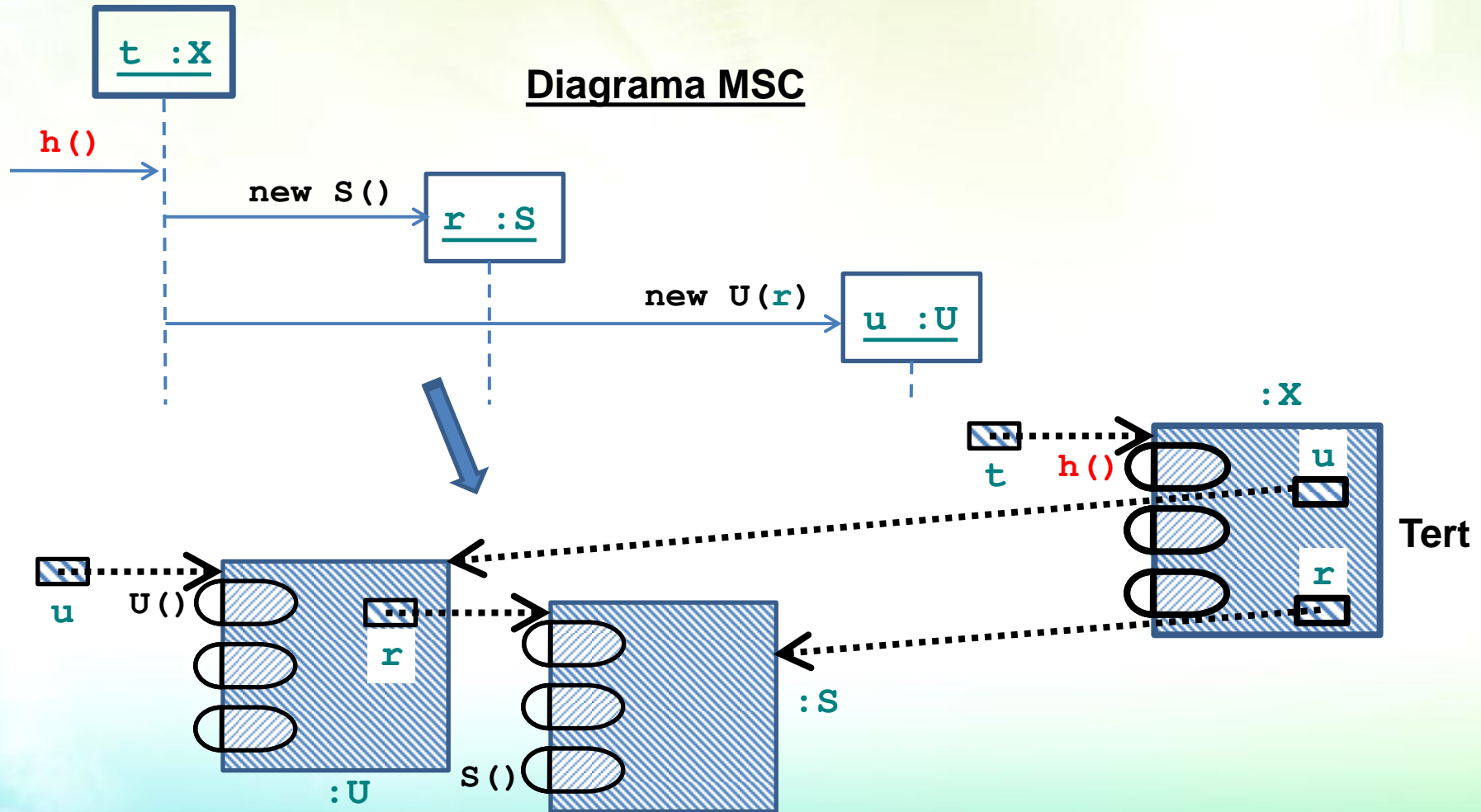
```
public class U { // clasa primului obiect
    private S r;
    public U(S o) { // primirea referintei catre al doilea obiect
        r = o; // stocarea referintei catre al doilea obiect
    }
}

public class S { // clasa celui de-al doilea obiect
    // constructor S() explicit sau implicit
}

public class X {
    private U u; // tertul are referinta catre primul obiect
    private S r; // tertul are referinta catre al doilea obiect
    ... h() {
        r = new S(); // crearea celui de-al doilea obiect
        u = new U(r); // crearea primului obiect si "legarea" lor
    }
}
```


Crearea obiectelor

2. Primul obiect il **primește** pe al doilea **in momentul crearii** de la un tert



Crearea obiectelor

3. Primul obiect il **primeste** pe al doilea **pe parcurs** de la un **tert**

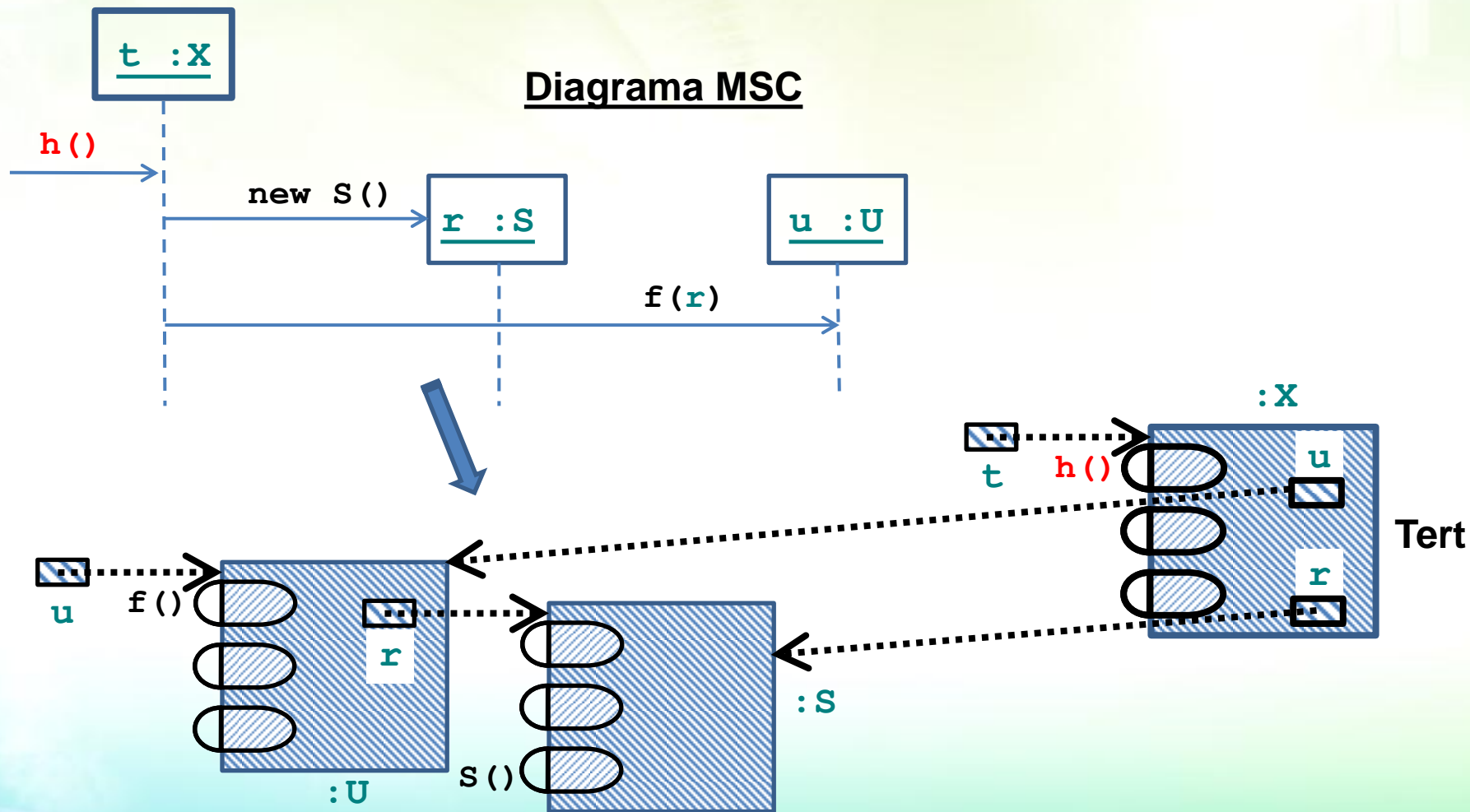
```
public class U { // clasa primului obiect
    private S r;
    ... f(S s) { // primirea referintei catre al doilea obiect
        r = s; // stocarea referintei catre al doilea obiect
    }
}

public class S { // clasa celui de-al doilea obiect
    // constructor S() explicit sau implicit
}

public class X {
    private U u; // tertul are referinta catre primul obiect
    private S r; // tertul are referinta catre al doilea obiect
    ... h() {
        r = new S(); // crearea celui de-al doilea obiect
        u.f(r); // "legarea" lui de primul obiect
    }
}
```

Crearea obiectelor

3. Primul obiect il **primește** pe al doilea **pe parcurs** de la un **tert**



Crearea obiectelor

4. Primul obiect il **obține** pe al doilea **pe parcurs** de la un tert

```
public class U { // clasa primului obiect
    private S r;
    private X t; // primul obiect are o referinta catre tert
    ... f() {
        r = t.g(); // obtinerea celui de-al doilea obiect
    }
}

public class S { // clasa celui de-al doilea obiect
    // constructor S() explicit sau implicit
}

public class X {
    private S r; // tertul are referinta catre al doilea obiect
    public S g() {
        r = new S(); // crearea celui de-al doilea obiect
        return r; // returnarea "legaturii" (referintei)
    }
}
```

Crearea obiectelor

4. Primul obiect il **obține** pe al doilea **pe parcurs** de la un tert

