

```
public class Creator {
    public static Prodotto getProdotto(String name) {
        switch (name) {
            case "primo": return new ProdottoA();
            case "secondo": return new ProdottoB();
            case "terzo": return new ProdottoC();
            case "quarto": return new ProdottoD();
            default: return new ProdottoA();
        }
    }
}
```

```
Prodotto p1 = Creator.getProdotto("primo");
```

Se usassimo un Supplier definito come

```
Supplier<Prodotto> prodSupplier = ProdottoA::new;
```

La linea di codice per creare p1 sarebbe tradotta in questo modo

```
Prodotto p1 = prodSupplier.get();
```

Quindi possiamo creare una mappa per far corrispondere al nome di un prodotto la sua creazione.

```
Map<String, Supplier<Prodotto>> map = Map.of("primo", ProdottoA::new, "secondo",
ProdottoB::new, "terzo", ProdottoC::new);
```

In questo modo abbiamo tradotto parte di switch. Adesso usiamo questa mappa per istanziare i prodotti:

```
public static Prodotto getProdotto(String name) {
    Supplier<Prodotto> s = map.get(name);
    if (s != null) return s.get();
    return new ProdottoA(); // Default
}
```

Qualora il costruttore avesse più parametri in ingresso non possiamo usare Supplier e dobbiamo usare un'altra interfaccia funzionale.