Builder

Angelo Afeltra

Intent

Separa la costruzione di un oggetto complesso dalla sua rappresentazione in modo che lo stesso processo di costruzione possa creare rappresentazioni diverse.

Problema

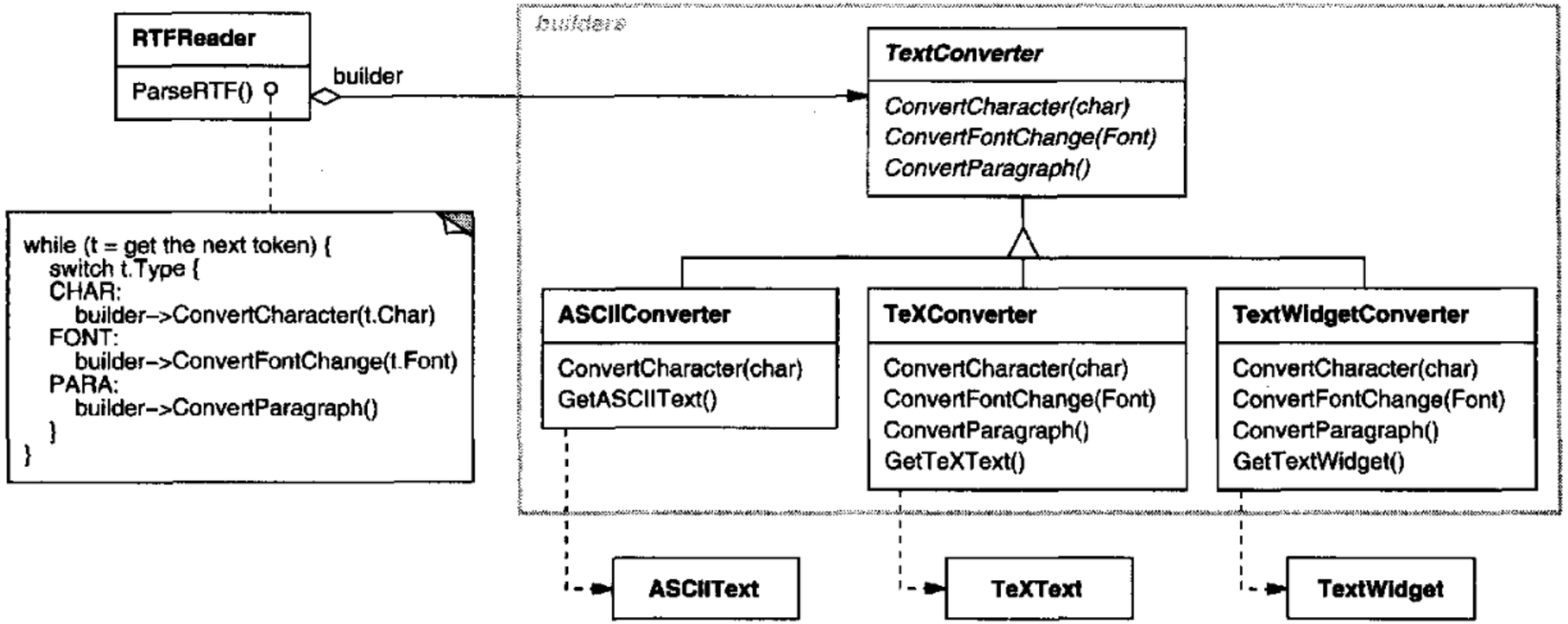
Supponiamo di dover sviluppare un lettore per la conversione di documenti RTF (Rich Text Format). Questo lettore deve avere la capacità di convertire tali documenti in vari formati, ad esempio trasformarli in testo ASCII standard o in un widget di testo che può essere modificato in modo interattivo.

Il problema principale è che il numero potenziale di conversioni è aperto, il che significa che dovremmo essere in grado di aggiungere nuove conversioni senza dover apportare modifiche al lettore originale.

Una soluzione possibile consiste nell'implementare la classe RTFReader con un oggetto chiamato TextConverter, il quale avrà il compito di convertire il documento RTF in diverse rappresentazioni testuali. In questo scenario, mentre RTFReader analizza il documento RTF, farà uso del TextConverter per eseguire le conversioni.

In questo modo, il TextConverter si occuperà sia dell'esecuzione delle conversioni dei dati che della rappresentazione dei token in un formato specifico. Le sottoclassi di TextConverter saranno specializzate in diverse conversioni e formati, permettendo di gestire l'aggiunta di nuove conversioni in modo flessibile senza dover modificare il lettore di base.

Motivazione

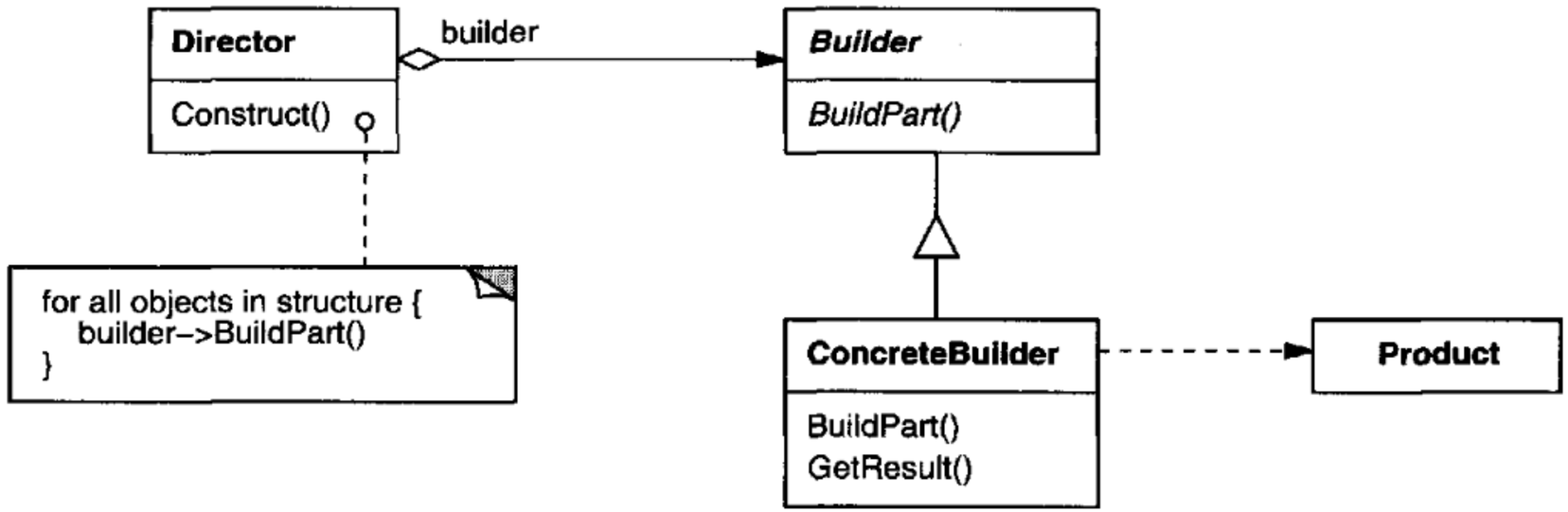


Quando usare il pattern Builder?

L'algoritmo per la creazione di un oggetto complesso dovrebbe essere indipendente dalle parti che compongono l'oggetto e da come sono assemblate.

Il processo di costruzione deve consentire rappresentazioni diverse per l'oggetto che viene costruito.

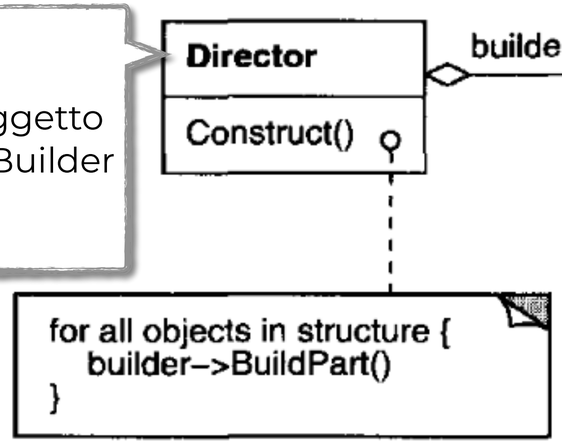
Struttura

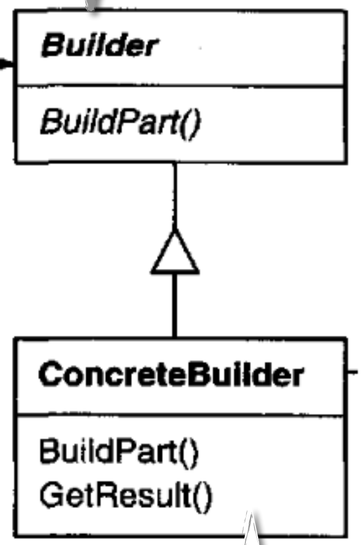

Director
builder
Construct()
Builder
BuildPart()
for all objects in structure {
builder->BuildPart()
}
ConcreteBuilder
BuildPart()
GetResult()
Product

Struttura

Specifica l’interfaccia astratta per creare parti di un Product Object

Costruisce un oggetto con l’interfaccia Builder





- Rappresenta l'oggetto complesso in costruzione. ConcreteBuilder costruisce la rappresentazione interna del prodotto e definisce il processo con cui è assemblato.
- Include classi che definiscono le parti costituenti, comprese le interfacce per assemblare le parti nel risultato finale.

- Costruisce e assembla parti del prodotto implementando l'interfaccia Builder.
- Definisce e tiene traccia della rappresentazione che crea.
- Fornisce un'interfaccia per il recupero del prodotto.

Conseguenze

PRO

Permette di variare la rappresentazione interna di un prodotto

Isola il codice per la costruzione e la rappresentazione

Ti da un controllo più preciso sul processo di costruzione

CONTRO

EMPTY

Conseguenze

PRO

Permette di variare la rappresentazione interna di un prodotto

Isola il codice per la costruzione e la rappresentazione

Ti da un controllo più preciso sul processo di costruzione

CONTRO

L'oggetto Builder fornisce al director un'interfaccia astratta per la costruzione del prodotto. L'interfaccia consente al costruttore di nascondere la rappresentazione e la struttura interna del prodotto. Nasconde anche come il prodotto viene assemblato. Poiché il prodotto è costruito attraverso un'interfaccia astratta, tutto ciò che devi fare per cambiare la rappresentazione interna del prodotto è definire un nuovo tipo di costruttore.

Conseguenze

PRO

Permette di variare la rappresentazione interna di un prodotto

Isola il codice per la costruzione e la rappresentazione

Ti da un controllo più preciso sul processo di costruzione

CONTRO

Il modello Builder migliora la modularità incapsulando il modo in cui un oggetto complesso è costruito e rappresentato. Ogni ConcreteBuilder contiene tutto il codice per creare e assemblare un particolare tipo di prodotto. Il codice viene scritto una volta; quindi diversi director possono riutilizzarlo per costruire varianti di prodotto dallo stesso set di parti.

Conseguenze

PRO

Permette di variare la rappresentazione interna di un prodotto

Isola il codice per la costruzione e la rappresentazione

Ti da un controllo più preciso sul processo di costruzione

CONTRO

Il pattern Builder costruisce il prodotto passo dopo passo sotto il controllo del director. Solo quando il prodotto è finito il director lo recupera dal costruttore. Quindi l'interfaccia di Builder riflette il processo di costruzione del prodotto più di altri modelli creational. Questo ti dà un controllo più preciso sul processo di costruzione e di conseguenza sulla struttura interna del prodotto risultante.

Intent

Separa la costruzione di un oggetto complesso dalla sua rappresentazione in modo che lo stesso processo di costruzione possa creare rappresentazioni diverse.

Problema

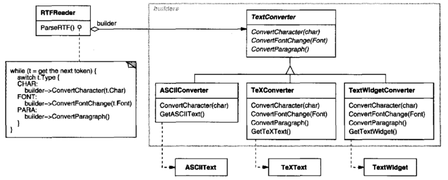
Supponiamo di dover sviluppare un lettore per la conversione di documenti RTF (Rich Text Format). Questo lettore deve avere la capacità di convertire tali documenti in vari formati, ad esempio trasformarli in testo ASCII standard o in un widget di testo che può essere modificato in modo interattivo.

Il problema principale è che il numero potenziale di conversioni è aperto, il che significa che dovremmo essere in grado di aggiungere nuove conversioni senza dover apportare modifiche al lettore originale.

Una soluzione possibile consiste nell'implementare la classe RTFReader con un oggetto chiamato TextConverter, il quale avrà il compito di convertire il documento RTF in diverse rappresentazioni testuali. In questo scenario, mentre RTFReader analizza il documento RTF, farà uso del TextConverter per eseguire le conversioni.

In questo modo, il TextConverter si occuperà sia dell'esecuzione delle conversioni dei dati che della rappresentazione dei token in un formato specifico. Le sottoclassi di TextConverter saranno specializzate in diverse conversioni e formati, permettendo di gestire l'aggiunta di nuove conversioni in modo flessibile senza dover modificare il lettore di base.

Motivazione

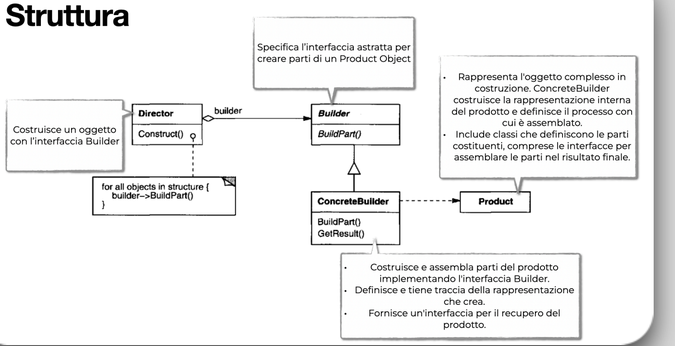


Quando usare il pattern Builder?

L'algoritmo per la creazione di un oggetto complesso dovrebbe essere indipendente dalle parti che compongono l'oggetto e da come sono assemblate.

Il processo di costruzione deve consentire rappresentazioni diverse per l'oggetto che viene costruito.

Struttura



Conseguenze

PRO

- Permette di variare la rappresentazione interna di un prodotto
- Isola il codice per la costruzione e la rappresentazione
- Ti da un controllo più preciso sul processo di costruzione

CONTRO

EMPTY

Builder