

Het Flyweight Pattern

Objectgeoriënteerde analyse en ontwerp

Het Flyweight Pattern

Doel

Efficiënt gebruik maken van een **grote hoeveelheid** zeer gelijkaardige **objecten**.

Voorbeeld



Het Flyweight Pattern

Het probleem dat het pattern aanpakt

Elk object heeft een toestand. Die toestand wordt in het geheugen bijgehouden.

Als je veel objecten hebt met elk veel toestand, dan gebruiken die veel geheugen. Dat is niet efficiënt.

Het flyweight pattern doet daar iets aan door gelijkaardige toestand van gelijkaardige objecten maar één keer op te slaan.

Het Flyweight Pattern

Intrinsieke en extrinsieke toestand

Een object heeft een **toestand**.

Een deel daarvan is **onafhankelijk van de context** (dus van de samenwerking met andere objecten). Dat is de **intrinsieke** toestand.

Het andere deel is wel **afhankelijk van de context**. Dat is de **extrinsieke** toestand.

Het Flyweight Pattern

Intrinsieke en extrinsieke toestand

Voorbeelden

De soldaten:

- Er zijn maar een paar soorten uitzicht: intrinsiek.
- Ze staan elk op een andere plaats: extrinsiek.

Lettertekens in een tekst:

- Het uitzicht van de letter is steeds hetzelfde: intrinsiek.
- De plaats van de letter in de tekst is voor elk teken verschillend: extrinsiek.

Flyweight en toestand

Flyweight beperkt het aantal objecten door de **intrinsieke toestand** te **delen**.

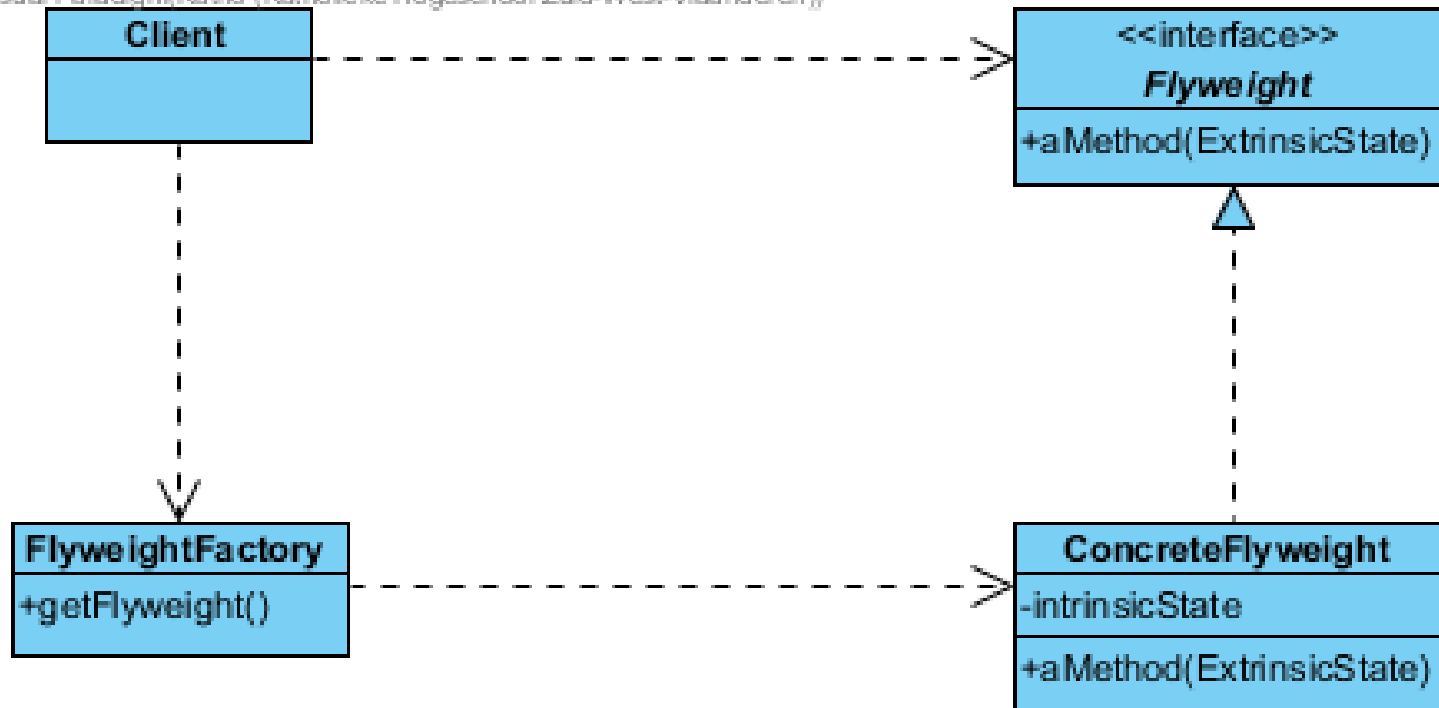
De **extrinsieke toestand** wordt **niet meer in het object** bewaard, maar wordt door de context aan het object bezorgd wanneer dat nodig is.

Clients mogen niet meer zelf objecten maken, maar moeten objecten vragen aan een **FlyweightFactory**.

De FlyweightFactory houdt een **pool** van flyweight-objecten bij.

Deelnemers

Visual Paradigm(Katho (Katholieke Hogeschool Zuid-West-Vlaanderen))



Deelnemers

- **Flyweight:** interface voor veel gebruikt object, dat gepoold wordt.
- **ConcreteFlyweight:** een concrete instantie van een Flyweight
- **FlyweightFactory:** houdt een pool van Flyweights bij, en stelt die ter beschikking van clients.
- **Client:** heeft een groot aantal flyweight objecten nodig, die geleverd worden door de flyweightfactory. De client moet de extrinsieke toestand van de flyweights bijhouden, en op het goede moment aan de flyweights bezorgen.

Welk soort objecten zijn geschikt voor Flyweight?

- 'Fine grained' objecten: dwz objecten met een beperkte verantwoordelijkheid, meestal low-level objecten,
- en waarvan er zo veel zijn dat ze te veel geheugen innemen,
- en waarvan een groot deel van de toestand extrinsiek gemaakt kan worden,
- en waarvan de resterende intrinsieke toestand een beperkt aantal waarden heeft.

Voor- en nadelen

Voordeel:

- Minder geheugen nodig

Nadelen:

- Bijkomende complexiteit door de factory
- Client moet extrinsieke toestand aan flyweight bezorgen
- Flyweights hebben geen identiteit, dus testen op identiteit werkt niet meer!