

P1W2 Generics – Opgave 1 "Generic Stack"

We willen zelf een verzamelingenklasse schrijven die generiek is (zoals het voorbeeld **Box** in de presentatie). De datastructuur die we gaan implementeren is een **LimitedStack**. Werk deze klasse uit volgens deze instructies (**de opgave bestaat uit 3 delen**):

Deel 1: LimitedStack

- Maak de klasse generiek, zodat ze kan gebruikt worden voor elk type. Pas dus de classheader aan.
- Maak de methode **push** generiek en werk ze verder uit: Voeg het nieuwe element toe aan de stack, maar test eerst op de capaciteit en doe een eventuele throw van een **FullStackException**.
- Maak de methode **pop** generiek en werk ze verder uit: Neem het meest recente element weg, maar test eerst op het aantal en doe een eventuele throw van een **EmptyStackException**.
- Maak de methode **top** generiek en werk ze verder uit: Retourneer het meest recente element, maar test eerst op het aantal en doe een eventuele throw van een **EmptyStackException**.
- Test nu uit door **DemoStack** te runnen.

Deel 2: LimitedNumericStack

- Maak een klasse **LimitedNumericStack** die overerft van **LimitedStack**. Pas op bij de toepassing van de type parameter van generics: je aanvaardt deze keer enkel elementen van de klasse **Number** en afgeleiden daarvan.
- Voorzie beide constructors
- Voorzie de methoden **push**, **pop** en **top**, maar roep telkens de superklasse op.
- De methoden **capacity**, **size** en **toString** moet je niet voorzien (overerving).
- Verwijder de commentaartekens in **DemoStack** en vergelijk jouw afdruk.

Uiteindelijke afdruk:

```
Stack is full; can not push Filip
Stack als string: [Boudewijn, Albert]
Bovenste element: Albert
capaciteit = 2 size = 2
Bovenaan wegnemen: Albert
Na pop: capaciteit = 2 size = 1
Stack als string: [Boudewijn, ?]
Stack als string: [1, 2, 3, ?, ?, ?, ?, ?, ?, ?]
Bovenste element: 3
capaciteit = 10 size = 3
Bovenaan wegnemen: 3
Na pop: capaciteit = 10 size = 2
Stack als string: [1, 2, ?, ?, ?, ?, ?, ?, ?, ?]
```

➔ Zie verder voor deel 3

Deel 3: MyNumericStack

- Maak een klasse **MyNumericStack** naar analogie met **LimitedNumericStack**. Maak nu gebruik van delegatie (of compositie) in plaats van overerving.
- Vervang de programmaregel

```
LimitedNumericStack<Integer> intStack = new LimitedNumericStack<>();
```

door

```
MyNumericStack<Integer> intStack = new MyNumericStack<>();
```

en test opnieuw uit.