

# Datastructuren 2023

## Programmeeropdracht 1: Stapel

Bouw vier versies van een stapel (ofwel *stack*) en een simpel testprogramma.

**Deadline.** Woensdag 27 september 23:59

### De ADT STAPEL

De abstracte datastructuur STAPEL heeft hier de volgende operaties:

- `void create()` — maak een lege stapel (zie hieronder)
- `bool isEmpty()` — bepaal of de stapel leeg is
- `void clear()` — maak de stapel leeg
- `bool push(newItem)` — voeg een `newItem` toe op de top van de stapel, en geef terug of de opdracht gelukt is
- `bool pop()` — verwijder het bovenste element van de stapel, en geef terug of de opdracht gelukt is
- `bool top(topItem)` — geef het bovenste stapelement, zonder dit te verwijderen, en geef terug of de opdracht gelukt is

Implementeer de ADT STAPEL in C++ op vier verschillende manieren:

1. met een ‘klassiek’ array,
2. met pointers, als gelinkte lijst,
3. met de STL `vector`,
4. rechtstreeks met de STL `stack`.

De eerste twee implementaties zijn al behandeld bij het college Programmeer-methoden, college 12, hoofdstuk Datastructuren (stapels, rijen en binaire bomen). Zie daar voor voorbeeld code.

Omdat in C++ de ADT functie `create` overeenkomt met de constructor hoeft `create` niet apart te worden geïmplementeerd.

Maak zinvol gebruik van een *class template* `<T>` zodat de stapel voor algemene type data beschikbaar is.

In theorie zou het mooi zijn als de specificaties (`.h`) en de implementaties (`.cc/.cpp`) gescheiden zouden kunnen blijven, maar dat laten we hier achterwege omdat de standaard compiler niet altijd goed kan omgaan met templates. *Dus*: maak vier `.h` files, één voor elke implementatie van de

stapel; deze header file bevat óók de code, niet alleen de headers van de functies.

## Test: MaxStapel

Test je code door een MaxStapel te implementeren, dit is een ‘gewone’ stapel die positieve integers bevat. Bovendien heeft de MaxStapel de bijzondere eigenschap dat op elk moment de hoogste waarde in de stapel gerapporteerd kan worden. Daar heb je twee ‘parallele’ gewone stapels voor nodig, omdat de elementen die dieper in de stapel liggen niet bekeken kunnen worden. De tweede stapel houdt op elk moment bovenop het maximum van de originele stapel.

De testfunctie leest als instructie een reeks integers. Elke positieve integer wordt gepusht op de stapel, voor `-1` wordt gepopt, voor `0` wordt de stapel geleegd, en bij `-2` wordt de maximale waarde uit de stapel gerapporteerd.

Bijvoorbeeld, als de string `5 -2 3 7 5 -2 -1 -1 -2 0 3 5 -2` ingevoerd wordt, is de uitvoer `5 7 5 5`.

Handel op een verstandige manier het gedrag bij een lege stapel af.

**Instructies.** Werk in tweetallen. Graag. Haal van de website het testprogramma `handoutmain23.cc` en breid dit uit om te laten zien dat je stack implementaties werken. De test-functie is *templated*, zodat elk van de stapel implementaties gekozen kan worden. Documenteer je code en vermeld de auteurs in elke file. Voor deze kleine opdracht is een verslag niet nodig.

Inleveren in groepen via BrightSpace. Mogelijkheid komt iets later beschikbaar (vanwege problemen met instellingen). Succes!