

C++ templates

Aufgabe 1: fixStack

Implementieren Sie eine Klasse `Stack`, die einen Stapel fester Größe (*fixStack*) **als array** implementiert, jedoch bei der Instanziierung quasi beliebige Datentypen für den Inhalt der Stackelemente akzeptiert (Template Klasse).

Operationen:

- Konstruktor
- `push` (liefert `false`, falls der Stack voll ist)
- `pop` (liefert `false`, falls der Stack leer ist)

Folgendes Testprogramm sollte dann nebenstehende Ausgabe erzeugen, wenn der Stapel **für maximal 10 Elemente** geschrieben wurde:

```
#include <iostream>
#include <iomanip>
#include "Stack.h"

using namespace std;

int main()
{
    Stack<int>  intStapel;
    Stack<char> charStapel;

    int i = 1;
    while(intStapel.push(i))
        charStapel.push((char)(64 + i++));

    char c;
    while(intStapel.pop(i))
    {
        charStapel.pop(c);
        cout << setfill(' ') << setw(2) << i
              << setfill('.') << setw(10) << c << endl;
    }
}
```

Ausgabe:

```
10.....J
 9.....I
 8.....H
 7.....G
 6.....F
 5.....E
 4.....D
 3.....C
 2.....B
 1.....A
```

Hinweise:

- Mit `enum { anzahl = 10 };` kann man sich einen Bezeichner verschaffen, der innerhalb der Klassendeklaration wie eine Konstante verwendet werden kann.
- Verwenden Sie unbedingt nur eine gemeinsame Datei für die Klassendeklaration und -definition (z.B. `Stack.h`).

Aufgabe 2: nixFixStack

In der Liste der template-Parameter sind auch Parameter **konkreter Typen** erlaubt.
Verwenden Sie einen solchen Parameter, um erst bei der Instanziierung die **Stackgröße** festzulegen.

Hier eine Anwendung und die Ausgabe:

```
#include <iostream>
#include <string>
#include "Stack.h"

int main()
{
    Stack<int, 5> int5Stapel;
    Stack<std::string, 6> string6Stapel;

    int i = 1;
    while(int5Stapel.push(i))
        i++;

    while(int5Stapel.pop(i))
        std::cout << i << std::endl;

    string6Stapel.push("!\\n");
    string6Stapel.push("Flensburg");
    string6Stapel.push("n ");
    string6Stapel.push("i ");
    string6Stapel.push("o ");
    string6Stapel.push("\\nM ");

    std::string s;
    while(string6Stapel.pop(s))
        std::cout << s;
}
```

Ausgabe:

```
5
4
3
2
1
M o i n
Flensburg!
```