1 Vector

Um die Move-Semantik, und somit effizientes Verschieben, von eigenen Klassen zu erlauben, müssen entsprechende Konstruktoren und Operatoren implementiert werden. In dieser Aufgabe sollen Sie eine primitive Alternative zum `std::vector` implementieren, welche nur die Methode `PushBack` zur Verfügung stellt. Zudem soll der Container nur mit der Klasse `Item` als Value-Elemente funktionieren und statisches Memory zur internen Speicherung verwenden. Der Header dieser Implementation sollte etwa folgendermassen aussehen:

vector.h

```
class Vector {
  static constexpr size_t kMaxMemory = 4;

 public:
  Vector();
  Vector(const Vector& other);

  void PushBack(const Item& item);

 private:
  Item memory_[kMaxMemory];
  size_t size_;
};
```

1.1 Aufgabe

a) Implementieren Sie die Klassen `Item` und `Vector` mit jeweils Copy-Konstruktor und erzeugen Sie darin jeweils ein Log auf die Konsole. Folgendes Hauptprogramm soll dann kompilieren und laufen gelassen werden können.

main.cpp

```
#include <utility>
#include <iostream>

#include "item.h"
#include "vector.h"

int main() {
  Vector vector_a;

  Item item_a;
  std::cout << "normal push" << std::endl;
  vector_a.PushBack(item_a);
```

```
13    std::cout << "move push" << std::endl;
14    vector_a.PushBack(std::move(item_a));
15    std::cout << "inplace push" << std::endl;
16    vector_a.PushBack(Item());
17
18    std::cout << "normal assign" << std::endl;
19    Vector vector_b = vector_a;
20
21    std::cout << "move assign" << std::endl;
22    Vector vector_c = std::move(vector_a);
23  }
```

b) Ergänzen Sie nun die beiden Klassen mit der Implementierung der Move-Semantik. Achten Sie dabei darauf, dass im Move des `Vector`s auch dessen verwaltete `Item`s verschoben werden sollen. Korrigieren Sie etwaige andere Unterschiede, bis Sie folgenden Output erhalten:

```
1  normal push
2    Item: copy-assign
3  move push
4    Item: move-assign
5  inplace push
6    Item: move-assign
7  normal assign
8    Vector: copy-ctor
9    Item: copy-assign
10   Item: copy-assign
11   Item: copy-assign
12 move assign
13   Vector: move-ctor
14   Item: move-assign
15   Item: move-assign
16   Item: move-assign
```