Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
#include <vector>
using std::vector;
// ————————— in main() —————————
// a 3D matrix
vector<vector<vector<int>*>>* pv_v_pv = ...;

pv_v_pv.at(0).at(0).at(0) = 7;             // (1)
(*pv_v_pv).at(0).at(0).at(0) = 7;          // (2)
(*(*pv_v_pv).at(0)).at(0).at(0) = 7;       // (3)
(*(pv_v_pv).at(0)).at(0).at(0) = 7;        // (4)
(*(*(*pv_v_pv).at(0)).at(0)).at(0) = 7;    // (5)
(*((*pv_v_pv).at(0)).at(0)).at(0) = 7;     // (6)
(*(*(pv_v_pv).at(0)).at(0)).at(0) = 7;     // (7)
```

Welche Zuweisung kompiliert?

- 1 (1)
- 2 (2)
- 3 (3)
- 4 (4)
- 5 (5)
- 6 (6)
- 7 (7)

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
#include <vector>
using std::vector;
// ———————— in main() ————————
// a 3D matrix
vector<vector<vector<int>*>>* pv_v_pv = ...;

pv_v_pv.at(0).at(0).at(0) = 7;               // (1)
(*pv_v_pv).at(0).at(0).at(0) = 7;            // (2)
(*(*pv_v_pv).at(0)).at(0).at(0) = 7;         // (3)
(*(pv_v_pv).at(0)).at(0).at(0) = 7;          // (4)
(*(*(*pv_v_pv).at(0)).at(0)).at(0) = 7;      // (5)
(*((*pv_v_pv).at(0)).at(0)).at(0) = 7;       // (6)
(*(*(pv_v_pv).at(0)).at(0)).at(0) = 7;       // (7)
```

Welche Zuweisung kompiliert?

1 (1)

2 (2)

3 (3)

4 (4)

5 (5)

6 (6) ●

7 (7)

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

!

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

!

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
pv_v_pv
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . . !

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
(*pv_v_pv)
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
(*pv_v_pv).at(0)
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
(*pv_v_pv).at(0)
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
((*pv_v_pv).at(0))
```

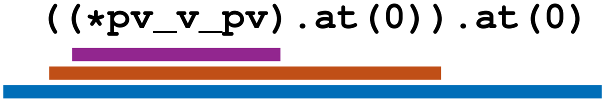
Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
((*pv_v_pv).at(0)).at(0)
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

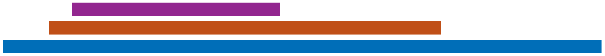


```
((*pv_v_pv).at(0)).at(0)
```

Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

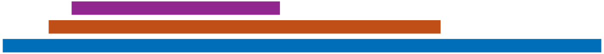
```
(*((*pv_v_pv).at(0)).at(0))
```



Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

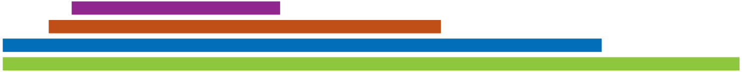
```
(*((*pv_v_pv).at(0)).at(0)).at(0)
```



Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

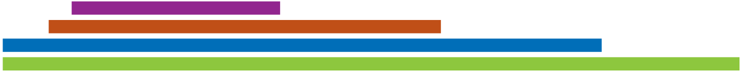
```
(*((*pv_v_pv).at(0)).at(0)).at(0)
```



Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
(*((*pv_v_pv).at(0)).at(0)).at(0) = 7
```



Zeiger, Vektoren, Vektoren voller Zeiger, . . .

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
vector<vector<int>*> v_pv = (*pv_v_pv).at(0);
vector<int>* pv = v_pv.at(0);
int& i = (*v_pv).at(0);

i = 7;
```

Schrittweises Vorgehen mittels lokaler Variablen kann dabei helfen, den Überblick über die Typen zu behalten.

Zeiger, Vektoren, Vektoren voller Zeiger, ...

```
vector<vector<vector<int>*>>* pv_v_pv = ...;
```

```
(*((*pv_v_pv).at(0)).at(0)).at(0) = 7;
```

```
pv_v_pv->at(0).at(0)->at(0) = 7;
```

Alternative Schreibweise `p->at(0)` statt `(*p).at(0)` resultiert in einem leichter lesbaren Ausdruck.