universität wien

Erweiterung der Klasse Vector

Dynamisches Wachstum

Diese Unterlagen sind verfügbar unter /home/Xchange/ue7/uebung7.pdf

Ziel

Die Klasse Vector soll bei Bedarf dynamisch wachsen.

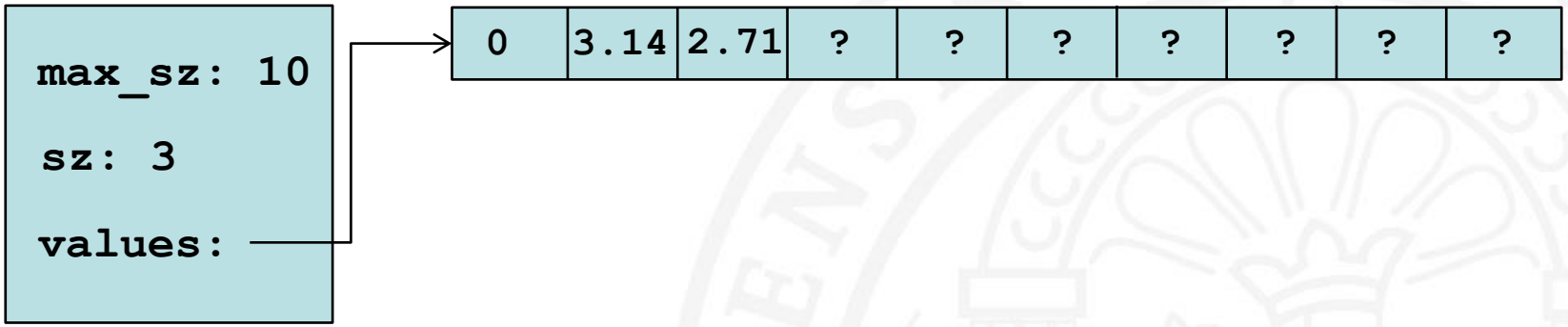
Wachstumsstrategie: Wenn für ein neu einzufügendes Element kein Platz mehr ist, so wird ein neuer Speicherblock mit doppelter Größe alloziert, die Inhalte des alten Blocks werden in den neuen Block kopiert und der alte Block wird freigegeben. Das einzufügende Element kann nun am nächsten verfügbaren Platz eingetragen werden.

Statt der konstanten maximalen Größe **`vector_max_size`** wird nun eine Instanzvariable **`max_sz`** verwendet, die die aktuelle Größe des allozierten Speicherbereichs angibt.

Statt eines statischen Arrays **`values`** wird ein Pointer **`values`** auf ein dynamisch alloziertes Array der Größe **`max_sz`** verwendet.

(Größe ist hier immer in Einheiten von double-Werten zu verstehen.)

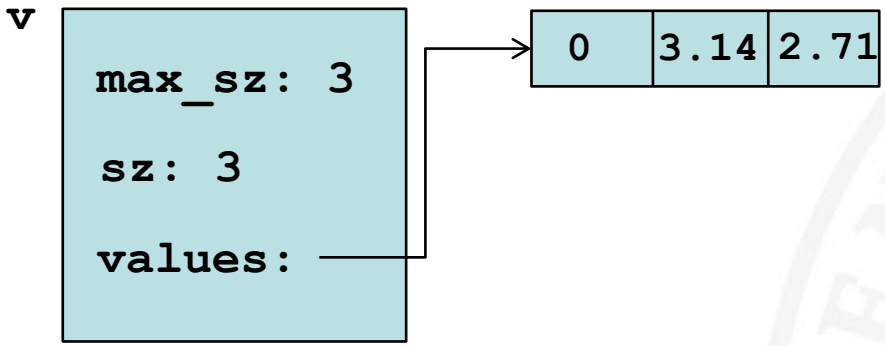
Ein dynamischer Vektor (schematisch)



Zu allen Zeitpunkten müssen folgende Integritätsbedingungen gelten:

- $0 \leq sz \leq max\_sz$
- $values$ ist ein dynamisch allozierter Speicherberich der Größe max_sz
- Die gespeicherten double Werte sind $values[0]$ ... $values[sz-1]$
- sz gibt den nächsten freien Index im Feld $values$ an.

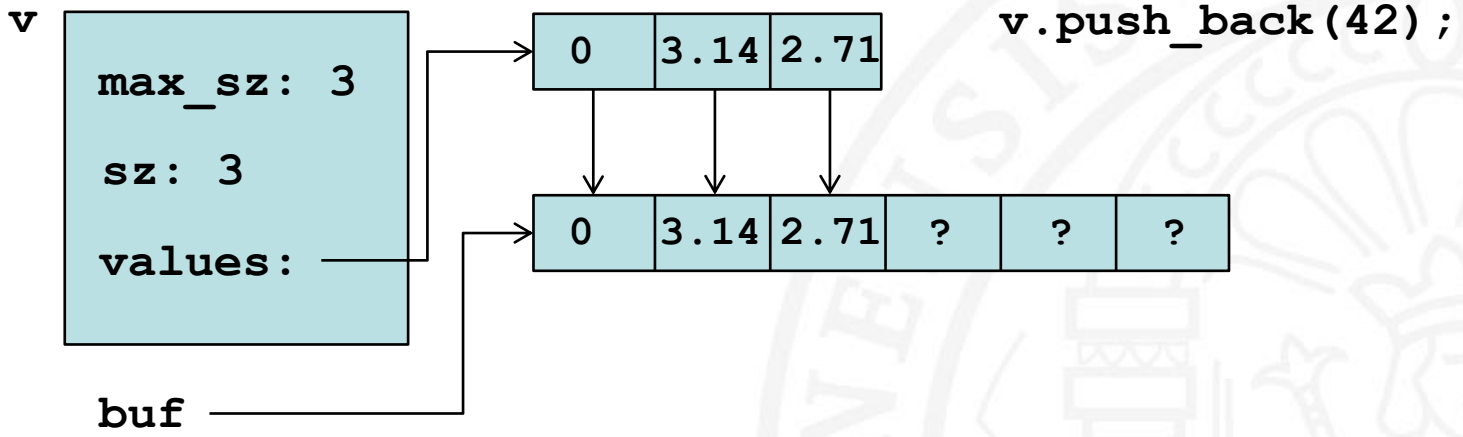
Einfügen und Vergrößern (1)




```
v.push_back(42);
```

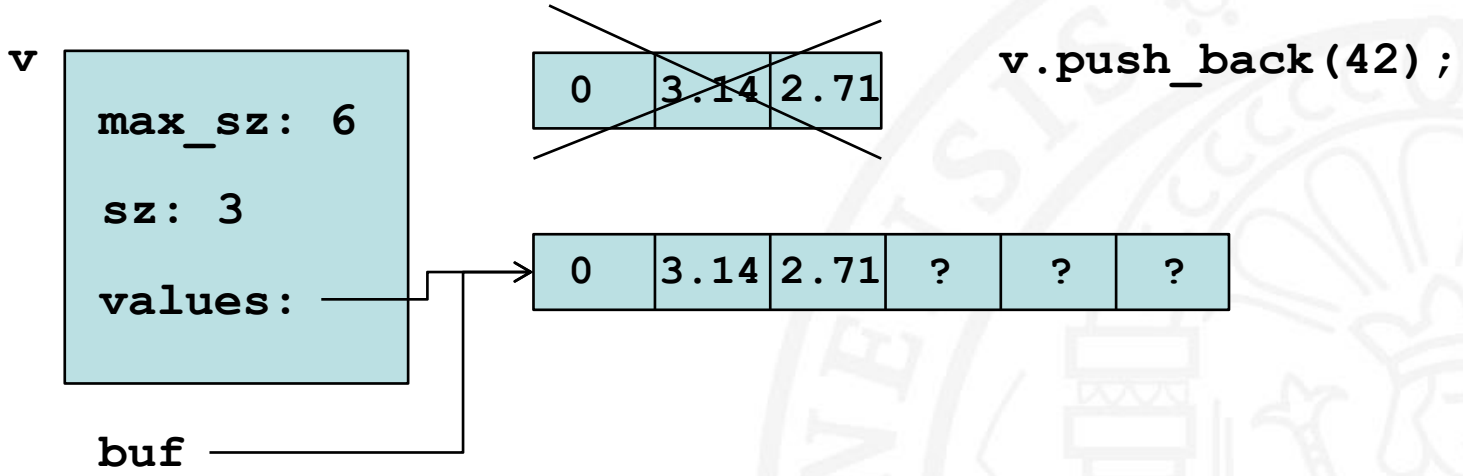
universität wien

Einfügen und Vergrößern (2)

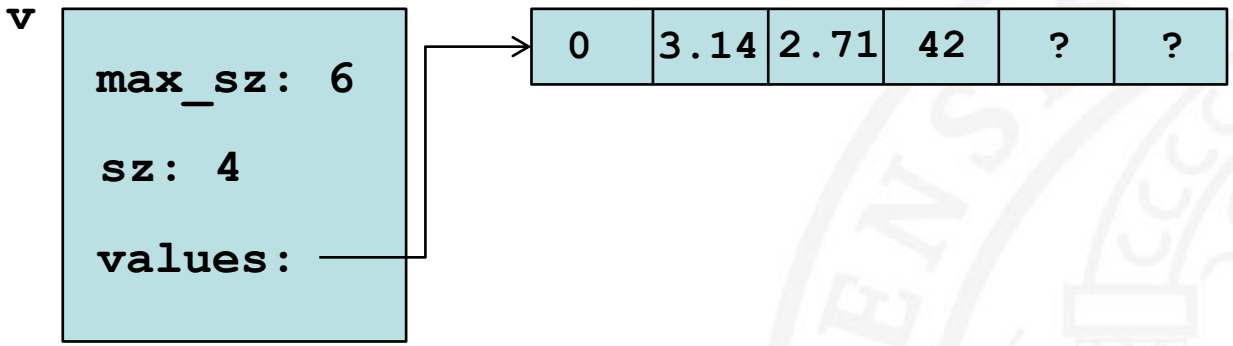


universität wien

Einfügen und Vergrößern (3)



Einfügen und Vergrößern (4)




```
v.push_back(42);
```

universität wien

Änderungen in der Klasse Vector

1) Instanzvariablen ändern / hinzufügen.
2) Neue Methoden implementieren / vorhandene Methoden anpassen (Da Arrays und Pointer verträglich sind, müssen nicht alle schon vorhandenen Methoden angepasst werden).
3) Zusätzliche Funktionalität implementieren.

universität wien

Methoden `reserve` und `shrink_to_fit`

Da das Vergrößern des Speicherbereichs an unterschiedlichen Stellen benötigt wird (**`push_back`**, **`insert`**), empfiehlt es sich, eine Methode dafür zu schreiben:

`void reserve(size_t)`: Vergrößert den Puffer, sodass mindestens so viele Elemente gespeichert werden können, wie der Parameter angibt (Eine Verkleinerung findet **nicht** statt).

Um überflüssigen Puffer wieder freigeben zu können, wird eine Methode **`void shrink_to_fit()`** implementiert, die den Puffer so weit reduziert, dass gerade noch die aktuell im Vektor gespeicherten Elemente Platz haben (falls eine minimale Anzahl von Elementen für die Klasse definiert wird, dann darf diese hier selbstverständlich nicht unterschritten werden).

universität wien

Methoden, die geändert werden müssen

- Konstruktoren: müssen Speicher für die Elemente reservieren. Entweder reserviert man immer zumindest eine minimale Anzahl an Elementen (diese kann in einer Klassenvariable vorgegeben werden) oder man lässt auch 0 für die Puffergröße zu (Das muss dann beim Vergrößern entsprechend berücksichtigt werden, da Verdoppeln von 0 nicht zu einer Vergrößerung führt).
- **`Vector(size_t)`**: Ein zusätzlicher Konstruktor, der es erlaubt, die zu reservierende Anzahl von Elementen im Parameter anzugeben.
- **`void push_back(double)`**: fügt einen Wert am Ende ein; **Vergrößern**, falls der Vektor schon voll ist (**keine Exception**).
- **`void insert(size_t, double)`**: fügt einen Wert an der vorgegebenen Position ein; Vergrößern, falls der Vektor schon voll ist.

Destruktor

- Es ist der Destruktor für die Klasse Vector zu implementieren. Dieser muss reservierten Speicher wieder freigeben.

universität wien

Weitere Funktionalität (globale Funktionen)

Überladen Sie **`operator>>`**, sodass Vektoren eingelesen werden können. Das Eingabeformat soll dabei dem Ausgabeformat (also z.B [1,2,3]) entsprechen. Was passiert, wenn der leere Vektor ([]) eingegeben wird? Können Sie das beheben?

Überladen Sie **`operator+`**, sodass zwei Vektoren miteinander mit + verknüpft werden können. Das Ergebnis der Operation soll ein Vektor sein, der alle Werte der beiden Vektor beinhaltet ("Zusammenhängen der beiden Vektoren" z.B.: [1,2,3]+[1,5,4] = [1,2,3,1,5,4]). Die beiden Operanden sollen nicht verändert werden.

Erstellen Sie eine Funktion **`Vector find(Vector srch, Vector pttrn)`**, die einen Vektor mit allen Indizes liefert, an denen der Vektor pttrn im Vektor srch auftritt.

z.B.: find([1,2,3,4,2,3,1,2,3], [2,3]) = [1,4,7], find([1,2,3],[7])=[]

Was sollte eine Suche nach einem leeren Vektor (find([1,2,3],[])) liefern?

Verwendung

Implementieren Sie eine Klasse Auto, bei welcher ein Vektor verwendet wird, um Tankvorgänge festzuhalten. Für ein Auto ist der Verbrauch (in l/100km) und der maximale Tankinhalt (in l) bekannt. Diese Größen sind beim Erstellen eines neuen Autos beim Konstruktoraufruf anzugeben. Außerdem soll ein Auto einen aktuellen Kilometerstand und einen aktuellen Tankinhalt haben. Beide sind zu Beginn mit 0 zu initialisieren. Der Vektor zur Speicherung der Tankvorgänge soll für ein neues Auto-Objekt leer sein. Schreiben Sie zwei Methoden void **`tanken(unsigned l)`** und void **`fahren(unsigned km)`**. Beide Methoden sollen Exceptions werfen, falls die jeweilige Aktion nicht vollständig durchgeführt werden kann. **`tanken`** erhöht den Tankinhalt, **`fahren`** vermindert ihn. **`tanken`** trägt (wenn es erfolgreich durchgführt werden kann) ein neues Element in den Vektor mit den Tankvorgängen ein.

Schreiben Sie weiters eine **`print`**-Methode und testen Sie Ihre Klasse.

Verwenden Sie Ihre Klasse Vector, um eine Klasse zu implementieren, die die aus der Mathematik bekannten Vektoren im $\mathbb{R}^n$ repräsentieren. Überladen Sie Operatoren um die üblichen Operationen (Addition, Subtraktion, Multiplikation mit einem Skalar und Skalarprodukt) mit solchen Vektoren ausführen zu können.

universität wien

Hinweis

Je nachdem, wie sorgfältig Sie Ihre Klassen testen, werden unter Umständen "seltsame" Fehler auftreten. Diese sind darauf zurückzuführen, dass für eine ordnungsgemäße Implementierung noch die Zuweisung und das Kopieren von Vektoren speziell behandelt werden müssen (das werden wir in der nächsten Übungseinheit korrigieren).