

5 Warteschlange und Stapel

- Warteschlange (queue): "First in First out" (FIFO)
- Stapel, Keller (stack): "Last in First out" (LIFO)

Die einfach verkettete Liste kann durch Auswahl der Einfüge- und Entnahmemethoden als Warteschlange oder Stapel benutzt werden.

vorneEinfügen, vorneEntnehmen

➔ Stapel

hintenEinfügen, vorneEntnehmen

➔ Warteschlange

vorneEinfügen, hintenEntnehmen

➔ Warteschlange

hintenEinfügen, hintenEntnehmen

➔ Stapel



Wie viele Züge sind nötig, um die Akten ABT, RUF, KUNZ alphabetisch sortiert herauszusuchen?

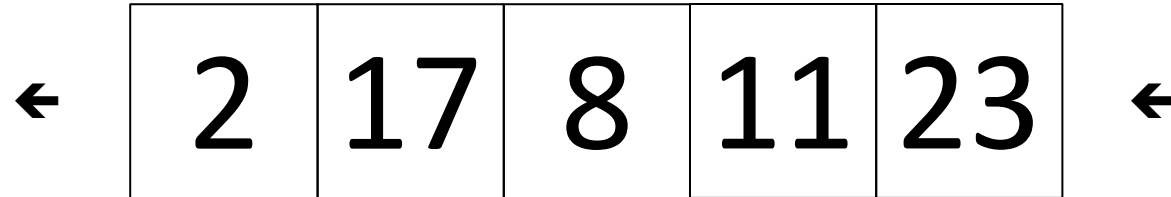
15 Züge



S. 34 / 1

S. 34 / 2

Warteschlange:



w.entnehmen()

w.entnehmen()

w.entnehmen()

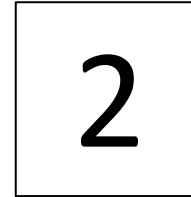
w.entnehmen()

w.entnehmen()

S. 34 / 1

S. 34 / 2

Warteschlange:



w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.einfuegen(2)

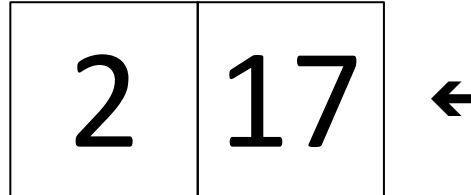
w.einfuegen(17)

S. 34 / 1

S. 34 / 2

Warteschlange:

←



w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.einfuegen(2)

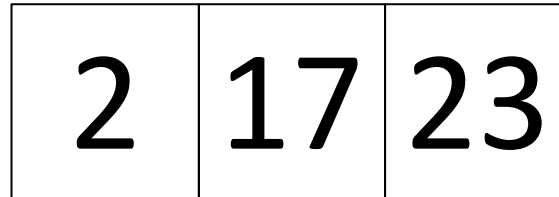
w.einfuegen(17)

w.einfuegen(23)

S. 34 / 1

S. 34 / 2

Warteschlange:



w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.einfuegen(2)

w.einfuegen(17)

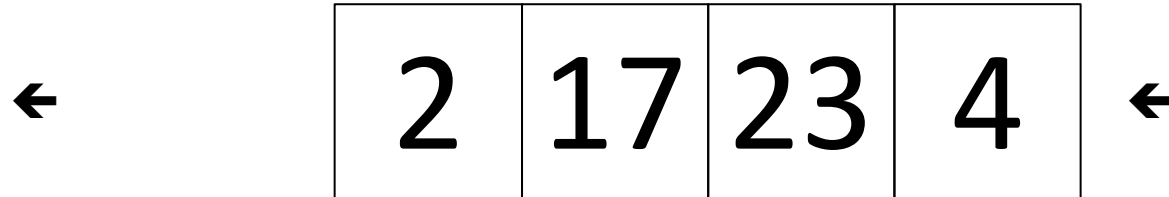
w.einfuegen(23)

w.einfuegen(4)

S. 34 / 1

S. 34 / 2

Warteschlange:



w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.entnehmen()

w.einfuegen(2)

w.einfuegen(17)

w.einfuegen(23)

w.einfuegen(4)

Stapel:



23	11	8	17	2
----	----	---	----	---

s.entnehmen()

s.entnehmen()

s.entnehmen()

s.einfuegen(23)

Stapel:



4	23	17	2
---	----	----	---

w.entnehmen()

w.entnehmen()

w.entnehmen()

s.einfuegen(23)

s.einfuegen(4)

Warteschlange:

Liste

- erster
- Liste()
- ~~• vorneEinfuegen(datenelement)~~
- vorneEntnehmen()
- ~~• hintenEinfuegen(datenelement)~~
- ~~• hintenEntnehmen()~~
- ~~• inhaltLetzterGeben()~~
- ~~• entfernen(datenwert)~~
- ~~• datenknotenGeben(datenwert)~~
- ~~• entnehmen(datenwert)~~
- anzahlElementeGeben()
- alleDatenAusgeben()
- istLeer()

<abstract>Listenelement

- naechsterGeben()
- inhaltGeben()
- hintenEinfuegen(datenelement)
- ~~• inhaltLetzterGeben(datenelement)~~
- ~~• entfernen(datenwert)~~
- ~~• datenknotenGeben(datenwert)~~
- anzahlDatenknotenGeben()
- listendatenAusgeben()

Datenknoten

- naechster

- inhalt

- Datenknoten(listenelement, datenelement)

- naechsterGeben()

- inhaltGeben()

- hintenEinfuegen(datenelement)

- ~~• inhaltLetzterGeben(datenelement)~~

- ~~• entfernen(datenwert)~~

- ~~• datenknotenGeben(datenwert)~~

- anzahlDatenknotenGeben()

- listendatenAusgeben()

Abschluss

- naechsterGeben()

- inhaltGeben()

- hintenEinfuegen(datenelement)

- ~~• inhaltLetzterGeben(datenelement)~~

- ~~• entfernen(datenwert)~~

- ~~• datenknotenGeben(datenwert)~~

- anzahlDatenknotenGeben()

- listendatenAusgeben()

Warteschlange Stapel

	Liste
● ●	erster
● ●	Liste()
●	vorneEinfuegen(datenelement)
● ●	vorneEntnehmen()
●	hintenEinfuegen(datenelement)
●	hintenEntnehmen()
	inhaltLetzterGeben()
	entfernen(datenwert)
	datenknotenGeben(datenwert)
	entnehmen(datenwert)
● ●	anzahlElementeGeben()
● ●	alleDatenAusgeben()
● ●	istLeer()

	<abstract>Listenelement
● ●	naechsterGeben()
● ●	inhaltGeben()
●	hintenEinfuegen(datenelement)
	inhaltLetzterGeben(datenelement)
	entfernen(datenwert)
	datenknotenGeben(datenwert)
● ●	anzahlDatenknotenGeben()
● ●	listendatenAusgeben()

Datenknoten

naechster

inhalt

Datenknoten(listenelement, datenelement)

naechsterGeben()

inhaltGeben()

hintenEinfuegen(datenelement)

inhaltLetzterGeben(datenelement)

entfernen(datenwert)

datenknotenGeben(datenwert)

anzahlDatenknotenGeben()

listendatenAusgeben()

Abschluss

naechsterGeben()

inhaltGeben()

hintenEinfuegen(datenelement)

inhaltLetzterGeben(datenelement)

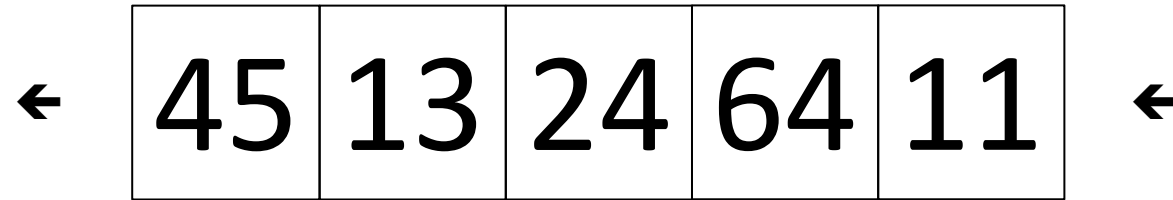
entfernen(datenwert)

datenknotenGeben(datenwert)

anzahlDatenknotenGeben()

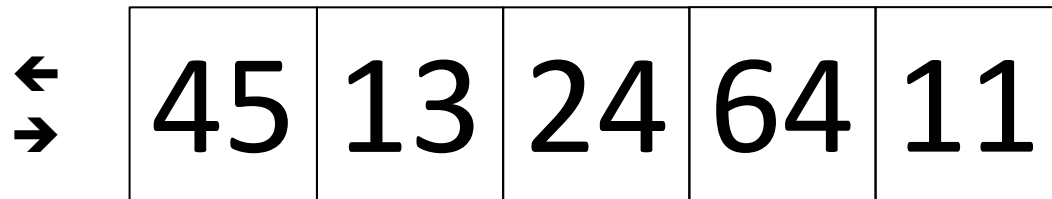
listendatenAusgeben()

Warteschlange:



Nr.24 wird storniert.

Stapel:



turmVerschieben(Anzahl n, Stapel quelle, Stapel hilf, stapel ziel):

start

wenn $n > 1$ dann

turmVerschieben(Anzahl $n-1$,
Stapel quelle, Stapel ziel, Stapel hilf)

wenn_ende

Versetze Scheibe von quelle nach ziel

wenn $n > 1$ dann

turmVerschieben(Anzahl $n-1$,
Stapel hilf, Stapel quelle, Stapel ziel)

wenn_ende

ende