

— Übungsblatt 6 (P) —

Aufgabe 17 (P)

(Mengen)

Schreiben Sie ein Programm, das unter Verwendung eines Set-Objekts die Ziehung der Lottozahlen simuliert. Lassen Sie dazu wiederholt eine ganzzahlige Zufallszahl im Zahlenbereich 1 bis 49 generieren und in das Set-Objekt einfügen, bis dieses genau sieben Elemente (einschließlich Zusatzzahl) enthält. Geben Sie dann die Elemente des Set-Objekts auf dem Bildschirm aus.

Aufgabe 18 (P)

(Mengen)

Schreiben Sie ein Programm, das unter Verwendung der Klasse `TreeSet` die Primzahlen zwischen 2 und n nach der folgenden Methode (Sieb des Eratosthenes) berechnet und ausdruckt:

1. Lies n ein.
 2. Erzeuge die Menge T als Menge aller Zahlen von 2 bis n .
 3. Erzeuge die zunächst leere Menge S .
 4. Setze p auf den Wert 2.
 5. Wiederhole
 - Nimm alle (nichttrivialen) Vielfachen von p (also $2p, 3p, \dots$) in S auf.
 - Falls p noch den Wert 2 hat, setze p auf 3, andernfalls erhöhe p so lange um 2, bis für p gilt $p \notin S$.
- so lange, bis gilt $p^2 > n$.
6. Entferne alle Elemente der Menge S aus der Menge T .
 7. Gib alle Elemente der Menge T aus (je 10 Werte in einer Zeile).

Aufgabe 19 (P)

(Schlangen)

Im Programm-Vorlagen-Verzeichnis auf der Moodle-Plattform stehen die Schnittstelle `WarteSchlange` sowie die Klassen `ZPatient` und `ZArztPraxis` zum Download für Sie bereit.

Die Klasse `ZArztPraxis` stellt eine Klassenmethode `warteZimmer` zur Verfügung, die menügesteuert die Vorgänge an der Patienten-Anmeldung einer Arzt-Praxis simuliert. Dabei wird mit Objekten der Klasse `ZPatient` und mit einer Warteliste in Form eines Objekts `wl`, das die Schnittstelle `WarteSchlange` implementiert, gearbeitet.

Das vorgegebene Interface `WarteSchlange` enthält lediglich die zwei (implizit abstrakten) Methoden

- `public void anmelden(ZPatient p)`
nimmt den Patienten `p` in die Warteliste auf.
- `public ZPatient derNaechsteBitte()`
liefert den jeweils nächsten Patienten und entfernt ihn aus der Warteliste.

Die Klasse `ZPatient` beinhaltet Instanzvariablen für den Namen und das Geburtsjahr des Patienten und für den Verweis auf den Nachfolger in einer Warteliste, einen Konstruktor, eine lesende Zugriffsmethode für das Alter und eine `toString`-Methode.

a) Schreiben Sie eine Klasse `FCFSListe`, die das Interface `WarteSchlange` implementiert und somit später als Parameter der Methode `ZArztPraxis.warteZimmer` verwendet werden kann. Realisieren Sie diese Liste nach dem Prinzip „First Come, First Serve“. Gehen Sie wie folgt vor:

- Statten Sie die Klasse mit zwei privaten Instanzvariablen vom Typ `ZPatient` aus, die jeweils eine Referenz auf den Anfang bzw. das Ende der Liste enthalten.
- Implementieren Sie die Methode `anmelden` so, dass neu anzumeldende Patienten immer am Ende der Liste eingefügt werden.
- Implementieren Sie die Methode `derNaechsteBitte` so, dass jeweils der Patient am Anfang der Liste als Ergebnis zurückgeliefert und aus der Liste entfernt wird.

b) Schreiben Sie ein kleines Testprogramm, das Ihre Implementierung einer Warteliste in Verbindung mit der Klasse `ZArztPraxis` testet. Erzeugen Sie dazu in Ihrer `main`-Methode ein Objekt der Klasse `FCFSListe` und verwenden Sie dieses Objekt als Parameter beim Aufruf der Klassenmethode `warteZimmer` der Klasse `ZArztPraxis`.

Ein Programmablauf sollte dann wie folgt aussehen:

```
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 1
Patienten-Aufnahme:  Name des Patienten: Jonathan Archer
                     Geburtsjahr: 2112
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 1
Patienten-Aufnahme:  Name des Patienten: Kathryn Janeway
                     Geburtsjahr: 2335
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 1
Patienten-Aufnahme:  Name des Patienten: William Riker
                     Geburtsjahr: 2335
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 2
Es kommt dran:  Jonathan Archer (geb. 2112)
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 1
Patienten-Aufnahme:  Name des Patienten: Jean-Luc Picard
                     Geburtsjahr: 2305
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 2
Es kommt dran:  Kathryn Janeway (geb. 2335)
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 2
Es kommt dran:  William Riker (geb. 2335)
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 2
Es kommt dran:  Jean-Luc Picard (geb. 2305)
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 2
Keine Patienten mehr im Wartezimmer!
Moegliche Aktionen: Patient anmelden (1), Patient drannehmen (2), Programmende (0)
Ihre Wahl: 0
```

Aufgabe 20 (P)

(Stapel)

Schreiben Sie eine Klasse `FeldStapel` (die Realisierung eines Stacks mit Hilfe einer Feld-Datenstruktur) und eine Klasse `KettenStapel` (die Realisierung eines Stacks mit Hilfe einer einfach verketteten Datenstruktur) sowie geeignete Testprogramme dazu. Vergleichen Sie den Aufwand für die typischen Stack-Operationen.