

1. Abstrakter Datentyp *Stack*:

- (a) Du kannst die für den Datentyp zentralen Methoden `push(element)`, `pop()`, `peek()`, `size()`, `isEmpty()` aufzählen und in Python auf der Grundlage einer Liste implementieren.
- (b) Du kannst die Kurzformel des Datentyps *Last In – First Out (LIFO)* angeben.
- (c) Du kannst den Verlauf von Operationen auf einem Stack nachvollziehen.
- (d) Du kannst mindestens drei verschiedene Anwendungen für Stacks aufzählen.
- (e) Du kannst in natürlicher Sprache einen Algorithmus beschreiben, der überprüft, ob die Klammern eines Ausdrucks korrekt gesetzt sind.
- (f) Du kannst arithmetische Ausdrücke von der Infix- in die Postfix-Notation umwandeln und umgekehrt.

2. Abstrakter Datentyp *Queue*:

- (a) Du kannst die für den Datentyp zentralen Methoden `enqueue(element)`, `dequeue()`, `size()` und `isEmpty()` aufzählen und in Python auf der Grundlage einer Liste implementieren.
- (b) Du kannst die Kurzformel des Datentyps *First In – First Out (FIFO)* angeben.
- (c) Du kannst mindestens drei verschiedene Anwendungen für Queues aufzählen.
- (d) Du kannst den Verlauf von Operationen auf einer Queue nachvollziehen.

3. Abstrakter Datentyp *Deque*:

- (a) Du kannst die Datenstruktur *Deque* beschreiben.
- (b) Du kannst die genaue Bedeutung der Abkürzung „*Deque*“ erklären.
- (c) Du kannst die für den Datentyp zentralen Methoden `addFront()`, `removeFront()`, `addRear()`, `removeRear()`, `size()`, `isEmpty()` aufzählen und in Python auf der Grundlage einer Liste implementieren.
- (d) Du kannst auf der Grundlage einer Deque eine Funktion schreiben, die überprüft, ob eine Zeichenkette ein Palindrom ist.

4. Abstrakter Datentyp *Linked List* (einfach verkettete Liste):

- (a) Du kannst die Datenstruktur einer *einfach verketteten Liste* aus einem *Head* und einer Folge von *Node*-Objekte graphisch darstellen (siehe Theorie, 2.4).
- (b) Du kannst die Elemente einer einfach verketteten Liste aus einem Speicherabbild herauslesen (siehe Aufgabe 3.1 [→ 2.11]).
- (c) Du kannst die für den Datentyp zentralen Methoden `add(item)`, `size()`, `remove(item)`, `search(item)`, `isEmpty()` aufzählen und die Laufzeitkomplexität $O(\dots)$ dieser Methoden angeben.

- (d) Du kannst das Einfügen eines Elements [`add(item)`] in die einfach verkettete Liste am graphischen Modell und am Speicherabbild darstellen (siehe Aufgabe 3.2 [→ 2.12]).
- (e) Du kannst das Entfernen eines Elements [`remove(item)`] aus der einfach verketteten Liste am graphischen Modell und am Speicherabbild darstellen (siehe Aufgabe 3.3 [→ 2.13]).
- (f) Du kannst die Bedeutung des Begriffs *Garbage Collection* in der Informatik erklären.