Mälardalens universitet

# DVA245, Laboration 2 – Circular Deque

| Mål |
| --- |
| **1- kunna använda abstrakta datatyper i programmeringsuppgifter** |
| **2 - kunna definiera och implementera abstrakta datatyper utifrån informella problembeskrivningar** |

Ni ska skriva en cirkulär deque – double-ended queue. Istället för funktionerna `insert` och `remove` som finns i `Queue`-klassen i `Queue.py` så ska er `Deque` ha funktionerna `insert_left, insert_right, remove_left` och `remove_right`.
Ni får utgå från `Queue` eller skapa en helt ny klass. Fundera på vilka medlemsvariabler som behövs. Fungerar samma namn som för `Queue`, eller behövs bättre namn? Vilka funktioner ovan motsvarar `insert` och `remove` i `Queue`? Vilka måste du lägga till som inte motsvaras av funktioner i `Queue`? Hur ser du till att indexen inte hamnar utanför listan? Hantera fallen där man försöker lägga till i en full deque eller ta bort ur en tom deque genom att lyfta felmeddelande (exception) som det är gjort i `Queue`.

Skriv tester för klassen, i en separat fil eller i en `main`-funktion i `Deque.py`. Sätt in och ta bort tillräckligt många värden från vänster och höger så att du testar alla möjligheter då dina medlemsvariabel-index skulle kunna hamna utanför listan. Det kan bli enklare att följa om du utökar `__str__` så att indexen för ändarna på kön finns med i strängen, eller ser till att de kan kontrolleras i testet på något annat sätt. Testa också att din klass lyfter felmeddelande (exception) om du försöker lägga till i en full deque eller ta bort ur en tom deque.

**För redovisning:**
Du ska kunna förklara hur testerna visar att din klass (funktionerna `insert_left, insert_right, remove_left` och `remove_right`) fungerar, och hur klassen fungerar.
Du ska också kunna förklara hur du kan använda din klass:

- Som en stack (vilka funktioner motsvarar push och pop)?
- Som en kö (vilka funktioner motsvarar enqueue/ insert och dequeue/ remove)?

**English:**

| **Learning outcomes** |
| --- |
| **1- be able to use abstract data types in programming assignments** |
| **2 - be able to define and implement abstract data types when given informal problem statements** |

You shall write a circular deque – double ended queue. In place of the functions `insert` and `remove` in the `Queue` class from `Queue.py` your `Deque` shall have the functions `insert_left` `insert_right`, `remove_left` and `remove_right`. You can start from `Queue` or write a completely new class. Think about what member variables are needed. Do the same names work as for the `Queue` class, or do you need better names? What functions above correspond to `insert` and `remove` in `Queue` ? What functions do you need to add? How do you make sure that indices are not outside the list? Handle the cases where you try to add to a full deque or remove from an empty deque by raising exceptions as in `Queue`.

Write tests for the class, in a separate file or in a `main`-function in `Deque.py`. Add and remove a sufficient amount of values from left and right so that you test all possibilities for the member indices to be outside of the list. It might be easier to follow if you extend `__str__` so that the indices of the deques ends are included in the string, or that you can control the indices in the tests in some other way. Also test that your class raises exception if you try to add to a full deque or remove from an empty deque.

**For presentation:**

You need to be able to explain how the tests show that the class (the functions `insert_left`, `insert_right`, `remove_left` and `remove_right`) work, and that the class works.

You shall also be able to explain how you can use your class:

- As a stack (what functions represent push and pop)?
- As a queue (what functions represent enqueue/ insert and dequeue/ remove)?