

Quilts

Cotton fabric used to be an expensive and scarce product. One dealt with it very economically: remainders or old pieces of clothing weren't thrown away, they were recycled. Old pieces were stitched together to serve as a protection against the cold. For example, soldiers in the Middle Ages wore quilted tunics over their clothing in order to stay warm. The Romans knew a kind of quilted bedding they called *culcita*. This word was later corrupted to *cowlte*, of which the word *quilt* was deducted.

Like so many workmanship and handiwork techniques, one made a virtue a necessity when quilting. Quilting a couple of layers of fabric became more and more decorative and was done by patterns, one more artistic than the other. If the upper layer of the quilt consisted of small, sown together pieces of fabric, it is called a *patchwork*-quilt. Today, the terms patchwork and quilt are both used to indicate the same thing, since most quilts are constructed from patchwork.

Today, there is an abundance of cheap cotton fabrics and there are affordable warm blankets and one of course no longer depends on remainders or old clothing. Making patchwork hasn't been a necessity for a while now, but it is still done as a hobby.



A modern patchwork-quilt.

Assignment

Define a class `Quilt` with which the patterns of a quilt can be represented. Every quilt corresponds with a representation of a number of characters in the format of a grid. For example, a 2×4 quilt could look as follows:

```
//-\n++||
```

The class must implement the following methods:

- An initializing method `__init__` to which three arguments must be given: *i*) the number of rows `m`, *ii*) the number of columns `n` and *iii*) a string. The quilt is filled out from left to right and from top to bottom with the consecutive characters from the given string. This string must have as many characters as the product of the number of rows `m` and the number of columns `n` that the quilt consists of. The initializing method must raise an `AssertionError` with the text `invalid configuration` if these conditions are not met. A quilt can only consist of valid characters, and the initializing method should also raise an `AssertionError` with the text `invalid configuration` if this should not be the case. By valid characters, we mean these symbols that we can still represent by a character from an ASCII character set if we rotate them 90 degrees: `\,/,+,*,-,|,o,x`.

- A method `__str__` that prints a string representation of the quilt. This string representation consists of `m` lines, where every line has `n` characters. The string representation itself does not end in a newline.
- A method `__repr__` that prints a string representation of the quilt. This string representation reads like a Python expression that makes a new object of the class `Quilt` has the same state.
- A method `rotate` that prints a new `Quilt` object, with a pattern that is rotated 90 degrees clockwise with regard to the pattern of the object on which this method is called.
- A method `__add__` that allows "sewing" two quilts together. Only quilts with an equal height can be sewn together. The method must raise an `AssertionError` with the text `quilts do not have equal height` if two quilts do not have an equal height. Otherwise the method must print a new `Quilt` object, of which the pattern consists of the patterns that were sewn together of the two objects that are summed up.

Objects of the class `Quilt` must be immutable. No other method may adjust the internal state of an object from the class `Quilt`. In other words, the internal state of the objects may not be changed after being initialized, which only allows making new objects.

Example

Remark: if you wish to use the interactive Python session below as a doctest, all backslashes of the output (that have already been escaped once below), must be escaped. In other words, all backslashes in the actual result must be replaced by a double backslash.

```
>>> quilt = Quilt(2, 2, '//++')
>>> print(quilt)
//
++
>>> quilt += Quilt(2, 2, '-\\|')
>>> print(quilt)
//-\
++||
>>> quilt
Quilt(2, 4, '//-\|++|')
>>> quilt = quilt.rotate()
>>> print(quilt)
+\
+\
-|
-|
>>> quilt
Quilt(4, 2, '+\\+\\-|/')
>>> quilt += Quilt(2, 2, '-\\|')
Traceback (most recent call last):
AssertionError: quilts do not have an equal height
>>> quilt = quilt.rotate()
>>> print(quilt)
||++
\//
>>> Quilt(2, 3, '++++')
Traceback (most recent call last):
AssertionError: invalid configuration
>>> Quilt(2, 3, 'oxoXo')
Traceback (most recent call last):
AssertionError: invalid configuration
```

Katoenen stof was in vroeger tijden een duur en schaars artikel. Men sprong er zuinig mee om: restanten of oude kledingstukken werden niet weggegooid, maar telkens opnieuw gebruikt. Oude lappen stikte men aan en op elkaar om te dienen als bescherming tegen de kou. Zo droegen soldaten in de Middeleeuwen doorgestikte tunieken over hun kleding om warm te blijven. De Romeinen kenden een soort gevoerd beddengoed, dat ze *culcita* noemden. Dit woord werd later verbasterd tot *cowlte*, waarvan het woord *quilt* is afgeleid.

Zoals met zoveel ambachtelijke en handwerktechnieken het geval is, maakte men ook bij het quilten van de nood een deugd. Het doorstikken van een paar lagen stof werd steeds decoratiever en gebeurde volgens patronen, de één nog kunstiger dan de andere. Bestond de bovenlaag van de quilt uit kleine, aan elkaar genaaide lapjes stof, dan spreekt men van een *patchwork*-quilt. De termen patchwork en quilt worden tegenwoordig vaak door elkaar gebruikt, omdat de meeste quilts samengesteld zijn uit patchwork.

Nu is er een overvloed aan goedkope katoenen stoffen en zijn er betaalbare warme dekens, en is men uiteraard niet meer afhankelijk van restjes of oude kleding. Het maken van patchwork is al lang geen noodzakelijke handenarbeid meer, maar wordt nog vaak uitgeoefend als hobby.



Een moderne patchwork-quilt.

Opgave

Definieer een klasse `Quilt` waarmee patronen van een quilt kunnen voorgesteld worden. Elke quilt correspondeert met een voorstelling van een aantal karakters in een roostervorm. Een 2×4 quilt zou er bijvoorbeeld als volgt kunnen uitzien:

```
//\n++||
```

De klasse moet volgende methoden implementeren:

- Een initialisatiemethode `__init__` waaraan drie argumenten moeten meegegeven worden: *i*) het aantal rijen `m`, *ii*) het aantal kolommen `n` en *iii*) een string. De quilt wordt van links naar rechts, en van boven naar onder opgevuld met de opeenvolgende karakters van de gegeven string. Deze string moet dus juist evenveel karakters tellen als het aantal rijen `m` vermenigvuld met het aantal kolommen `n` waaruit de quilt bestaat. De initialisatiemethode moet een `AssertionError` met de tekst `ongeldige configuratie` opwerpen indien niet aan deze voorwaarden voldaan is. Een quilt kan enkel uit geldige karakters bestaan, en de initialisatiemethode moet eveneens een `AssertionError` met de tekst `ongeldige configuratie` opwerpen als dat niet het geval is. Onder geldige karakters verstaan we deze symbolen die we na 90 graden draaien nog steeds door een karakter uit de ASCII tekenset kunnen voorstellen: `\,/,+,*,-,|,0,x`.
- Een methode `__str__` die een stringvoorstelling van de quilt teruggeeft. Deze

stringvoorstelling bestaat uit m regels, waarbij elke regel n karakters telt. De stringvoorstelling eindigt zelf niet op een newline.

- Een methode `__repr__` die een stringvoorstelling van de quilt teruggeeft. Deze stringvoorstelling leest als een Python expressie die een nieuw object aanmaakt van de klasse `Quilt` dat dezelfde toestand heeft.
- Een methode `draai` die een nieuw `Quilt` object teruggeeft, met een patroon dat 90 graden in wijzerzin gedraaid is ten opzichte van het patroon van het object waarop deze methode wordt aangeroepen.
- Een methode `__add__` die toelaat twee quilts aan elkaar te "naaien". Enkel quilts met dezelfde hoogte kunnen aan elkaar genaaid worden. De methode moet een `AssertionError` met de tekst `quilts zijn niet even hoog` opwerpen als de twee quilts ongelijke hoogtes hebben. Anders moet de methode een nieuw `Quilt` object teruggeven, waarvan het patroon bestaat uit de aan elkaar genaaide patronen van de twee objecten die bij elkaar opgeteld worden.

Objecten van de klasse `Quilt` moeten onveranderlijk (*immutable*) zijn. Geen enkele methode mag de interne toestand van een object van de klasse `Quilt` wijzigen. Met andere woorden, de interne toestand van de objecten mag niet meer gewijzigd worden nadat de objecten geïntialiseerd zijn, waardoor het enkel mogelijk is om nieuwe objecten aan te maken.

Voorbeeld

Opmerking: als je de onderstaande interactieve Python sessie wilt gebruiken als doctest, dan moeten alle backslashes van de uitvoer (die hieronder al een keer ge-escaped zijn) ge-escaped worden. Met andere woorden, alle backslashes in het eigenlijke resultaat moeten dus vervangen worden door een dubbele backslash.

```
>>> quilt = Quilt(2, 2, '//++')
>>> print(quilt)
//
++
>>> quilt += Quilt(2, 2, '-\\|')
>>> print(quilt)
//-\
++||
>>> quilt
Quilt(2, 4, '//-\|++||')
>>> quilt = quilt.draai()
>>> print(quilt)
+\
+\
-|
-/
>>> quilt
Quilt(4, 2, '+\\+\\-|/')
>>> quilt += Quilt(2, 2, '-\\|')
Traceback (most recent call last):
AssertionError: quilts zijn niet even hoog
>>> quilt = quilt.draai()
>>> print(quilt)
||++
\//
>>> Quilt(2, 3, '++++')
Traceback (most recent call last):
AssertionError: ongeldige configuratie
```

```
>>> Quilt(2, 3, 'oxooXo')
```

```
Traceback (most recent call last):
```

```
AssertionError: ongeldige configuratie
```