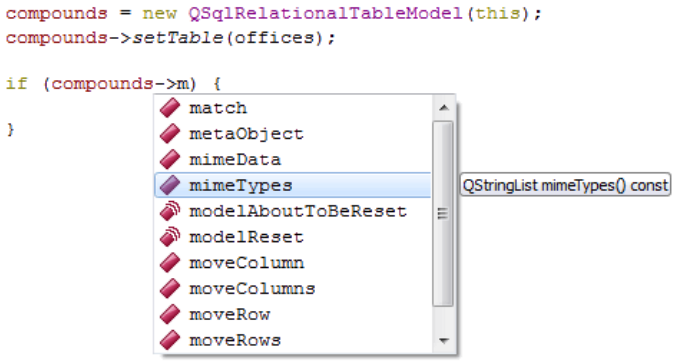
Autocomplete

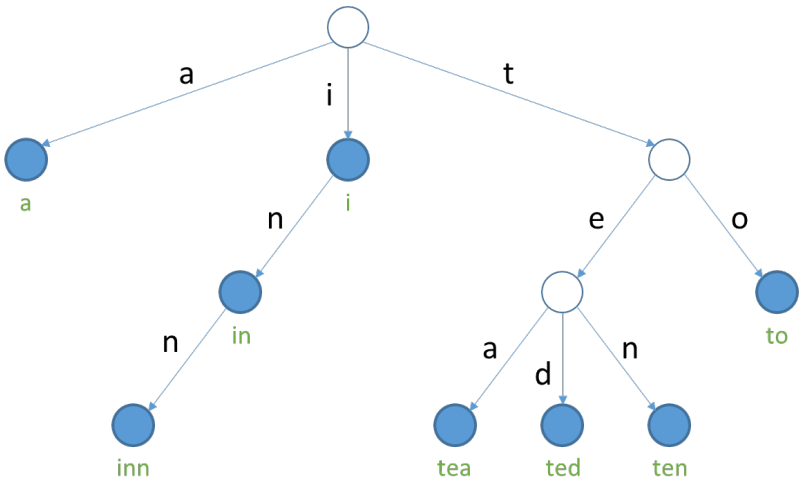
Automatische aanvulling — ook wel **auto-aanvullen** of **autocomplete** genoemd — is een optie die door veel broncode-editors, tekstverwerkers en webbrowsers wordt aangeboden. Daarbij probeert de software te voorspellen welk woord de gebruiker wil typen, zonder dat de gebruiker het al volledig heeft ingetikt. Deze aanpak kan snelheidswinst opleveren wanneer het relatief eenvoudig is om nieuw ingevoerde woorden te voorspellen *i*) gebaseerd op de woorden die al zijn ingevoerd, *ii*) omdat een beperkt aantal woorden typisch zijn voor e-mailprogramma's, browsers, shells van besturingssystemen, of *iii*) omdat de taal sterk gestructureerd en makkelijk te voorspellen is zoals in broncode-editors. Daardoor maakt automatische aanvulling de interactie met de computer sneller en aangenamer.



Autocomplete in Qt Creator 5.0: De programmeur typt code in. Als de software een bepaalde string herkent als de naam van een variabele, een methode of een klasse, dan wordt een menu weergegeven dat de volledige namen van variabelen, methoden of klassen bevat. De programmeur kan hieruit zijn keuze maken aan de hand van de muis of via de pijltjestoetsen. Als de programmeur blijft verdertypen zonder een keuze te maken, dan verdwijnt het menu.

Opgave

Om snel alle woorden te kunnen opzoeken die beginnen met een gegeven prefix, zullen we onderstaande zoekboom implementeren. In dit voorbeeld bevat de zoekboom de woorden a, i, in, inn, tea, ted, ten en to.



Zoekboom die de woorden a, i, in, inn, tea, ted, ten en to bevat. De accepterende knopen worden in het blauw opgevuld, en geven aan welke knopen een geassocieerde string hebben die

overeenkomt met een woord dat in het boom bevat zit.

Elke boog naar een knoop van de zoekboom is gelabeld met één enkel karakter. De labels van alle bogen die vertrekken uit eenzelfde knoop zijn allemaal verschillend. Door alle karakters op het pad van de wortel naar een knoop achter elkaar te zetten, krijg je een string die met die knoop geassocieerd is. De wortel van de zoekboom is geassocieerd met de lege string.

Slechts een deelverzameling van de knopen in de zoekboom hebben een geassocieerde string die correspondeert met een woord dat in de zoekboom bevat zit. Deze worden *accepterende knopen* genoemd, en worden in bovenstaande figuur in het blauw opgevuld. De niet-accepterende knopen worden in bovenstaande figuur in het wit opgevuld. Alle accepterende knopen in de deelboom van een knoop hebben de string die met die knoop geassocieerd is als gemeenschappelijke prefix.

Om een nieuw woord aan de zoekboom toe te voegen, overlopen we één voor één de karakters van het woord en volgen daarbij het pad vanaf de wortel dat wordt aangegeven door de labels van de bogen. Dit proces stopt als we een knoop bereikt hebben van waaruit geen boog vertrekt die gelabeld is met het volgende karakter van het woord (of als we alle karakters van het woord overlopen hebben). De knoop die we op deze manier bereiken, is geassocieerd met de langste gemeenschappelijke prefix van het toe te voegen woord en de woorden die reeds in de zoekboom bevat zitten. Daarna maken we voor de resterende karakters van het woord telkens een nieuwe boog die leidt naar een nieuwe knoop. De laatste knoop die in dit proces bereikt wordt (dit kan zowel een bestaande als een nieuw aangemaakte knoop zijn) is geassocieerd met het toegevoegde woord, en wordt dus gemarkeerd als een accepterende knoop.

Het zoeken naar woorden in de zoekboom gebeurt analoog als het toevoegen van woorden. Daarbij wordt enkel aangegeven dat het woord voorkomt in de zoekboom, als we na het doorlopen van alle karakters van het woord in een accepterende knoop terechtkomen.

Gevraagd wordt om een klasse `Autocomplete` te definiëren die een zoekboom implementeert conform aan bovenstaande beschrijving. Deze klasse moet minstens de volgende methoden ondersteunen:

- Een initialisatiemethode waaraan nul of meer woorden als afzonderlijke argumenten kunnen doorgegeven worden. De opgegeven woorden moeten aan de zoekboom toegevoegd worden bij initialisatie van het object.
- Een methode `add` waaraan een woord moet doorgegeven worden dat aan de zoekboom moet toegevoegd worden. De methode moet een verwijzing naar het object zelf teruggeven, zodat methode-aanroepen aan elkaar kunnen geschakeld worden.
- Een methode `extend` waaraan een reeks (een lijst of een tuple) van woorden moet doorgegeven worden, die individueel aan de zoekboom moeten toegevoegd worden. De methode moet een verwijzing naar het object zelf teruggeven, zodat methode-aanroepen aan elkaar kunnen geschakeld worden.
- Een methode `isPrefix` waaraan een string moet doorgegeven worden. De methode moet een Booleaanse waarde teruggeven die aangeeft of de zoekboom woorden bevat met de opgegeven prefix.
- Een methode `iteritems` waaraan een string moet doorgegeven worden. De methode moet een iterator teruggeven die alle woorden uit de boom die beginnen met de opgegeven prefix overloopt, gesorteerd in oplopende volgorde (volgorde zoals gedefinieerd op strings).
- Een methode `items` waaraan een string moet doorgegeven worden. De methode moet een

lijst teruggeven met alle woorden uit de boom die beginnen met de opgegeven prefix, gesorteerd in oplopende volgorde (volgorde zoals gedefinieerd op strings).

Zorg ervoor dat de operator `in` kan gebruikt worden om na te gaan of een gegeven woord al dan niet voorkomt in de zoekboom.

Voorbeeld

```
>>> boom1 = Autocomplete()
>>> wortel = boom1.add('a').add('i').add('tea')
>>> 'tea' in boom1
True
>>> 'te' in boom1
False
>>> wortel = boom1.extend(['in', 'inn', 'ted', 'ten', 'to'])
>>> 'ten' in boom1
True
>>> 'te' in boom1
False

>>> boom2 = Autocomplete('maandag', 'dinsdag', 'woensdag')
>>> 'maandag' in boom2
True
>>> 'donderdag' in boom2
False
>>> wortel = boom2.add('donderdag').add('vrijdag')
>>> 'donderdag' in boom2
True
>>> 'zaterdag' in boom2
False
>>> wortel = boom2.extend(('zaterdag', 'zondag'))
>>> 'zaterdag' in boom2
True
>>> 'kerstdag' in boom2
False
>>> boom2.isPrefix('maan')
True
>>> boom2.isPrefix('kerst')
False
>>> tuple(boom2.iteritems('z'))
('zaterdag', 'zondag')
>>> boom2.items('d')
['dinsdag', 'donderdag']
```