

VOORBEELD: Toets: Helpdesk

Bij een groot bedrijf kunnen alle werknemers (employees) hulp krijgen van de Informatiseringshelpdesk voor problemen die te maken hebben met de IT hulpmiddelen die geïnstalleerd zijn op de werkplek. Deze problemen kunnen te maken hebben met hardware of software.



In het systeem worden medewerkers bijgehouden. Een medewerker heeft een username en een password (dat gewoon in tekst wordt opgeslagen). Er zijn reguliere medewerkers (die meldingen mogen maken) en er zijn helpdeskmedewerkers (die meldingen mogen oplossen). Bij alle medewerkers wordt bijgehouden op welke afdeling ze werken. Voor helpdeskmedewerkers wordt ook nog bijgehouden wat hun expertise is.

Iedereen kan tickets toevoegen aan het systeem (Dat gebeurt in deze toets gewoon door C++ code toe te voegen aan de main methode). Er wordt dan een ticket met een uniek nummer aangemaakt. De datum van de melding worden vastgelegd in het systeem. De melder wordt ook vastgelegd. Daarnaast bevat het ticket een omschrijving van het probleem. Bij een hardware probleem moet ook de id van de computer worden vastgelegd, dat is een getal. Bij een softwareprobleem moet de naam van het softwarepakket ook worden opgeslagen.

Tickets kunnen opgelost worden door helpdeskmedewerkers. Als die helpdeskmedewerker een probleem oplost wordt opgeslagen welke medewerker dit was en kan er een korte oplossing voor het probleem beschreven worden.

Let op: Je hoeft geen menu of andere userinterface te maken voor het systeem. Alle medewerkers die het systeem gaan gebruiken kunnen programmeren en het volstaat om een *main* te maken waarin je geschreven methodes aanroept (zie het bijgeleverde main.cpp bestand voor details).

Let op: Check ook de achterkant!

De stappen om het systeem te bouwen

- Ontwerp het systeem op kladpapier. Bedenk alle classes en hun attributen en methodes. Teken de meest optimale oplossing (we kunnen alvast verklappen dat je overerving daarbij nodig hebt). Pas zoveel mogelijk alles toe wat je geleerd hebt. Je hoeft dit ontwerp niet in te leveren.
- Lees main.cpp aandachtig door en kijk wat je systeem uiteindelijk moeten kunnen doen. Eén van de onderdelen is het inlezen van een bestand. Dit bestand is bijgeleverd en mag niet aangepast worden, je kunt het bestand openen met de bestandsnaam 'input.txt'.
- Schrijf de benodigde classes. Zorg voor goede output() methodes voor elke class, deze methode schrijft de class naar 'std::cout'.
- Schrijf de helpdesk class.
- **Let op:** Zorg ervoor dat aan het einde van het programma ook alle classes zijn opgeruimd en alle geheugen is vrijgegeven. Aan het eind van de applicatie moet dus al het geheugen dat je hebt gealloceerd ook weer zijn vrijgegeven.
- **Let op:** Iedere class **moet** gedeclareerd worden in zijn eigen header file en iedere implementatie van de class in zijn eigen cpp file. Je mag dus niet alle code in de main.cpp zetten, je moet de classes opsplitsen over verschillende source en header files.

Enkele tips

Omdat we begrijpen dat best lastig kan zijn om een dergelijk systeem te bouwen hebben we hier enkele tips:

- Test tijdens het bouwen voortdurend. Dit kun je doen door aanroepen vanuit je *main* te doen naar je administratie class en eventueel wat te printen. Zorg wel dat je aan het einde alleen alle stappen doet die aangegeven zijn in het commentaar in de main, zodat je onderstaande uitvoer krijgt.
- Zijn er delen waar je niet uit komt? Probeer er dan omheen te werken zodat je toch nog punten kunt scoren op de andere onderdelen. Beschrijf eventueel in commentaar regels wat je probeert te doen, zodat de beoordelaar snapt waarom je bepaalde delen wel of niet gedaan hebt.
- Voor de duidelijkheid. We beoordelen op:
 - o De werking van je programma en de netheid van je code
 - o Het verdelen van de verantwoordelijkheden over de classes, je individuele classes en het gebruik van overerving en access-specifiers
 - o Het correct gebruik van header en cpp files en het uitsplitsen van classes over verschillende source file
 - o Het gebruik van STL datastructuren en algoritmen
 - o Correct gebruik en afhandelen van excepties
 - o Correcte allocatie en deallocatie van geheugen
 - o Correct gebruik van method/functie argumenten (by-value-, by-reference, by-address) en gebruik van const methoden waar mogelijk

Voorbeelduitvoer

Uitvoer van het systeem zou er als volgt uit kunnen zien:

Employees:

```
ruud (IT Lecturers)
jeroen (IT Lecturers)
hesther (Management)
gerralt (Helpdesk, Expertise: Mice)
```

Tickets:

Ticket 1 (Softwareticket, Application: Office 365):

Status: Open

ruud (IT Lecturers): Excel can't calculate the number of days in a year

Ticket 2 (Hardwareticket, computerId: 5):

Status: Solved

jeroen (IT Lecturers): My mouse doesn't work!

gerralt (Helpdesk, Expertise: Mice): Try putting it into the USB port instead of the HDMI port.

Ticket 3 (Softwareticket, Application: MacOS X):

Status: Solved

ruud (IT Lecturers): MacOS won't boot on my system

gerralt (Helpdesk, Expertise: Mice): Please replace your Dell laptop with a real system