**PRC31**

**Programming in C++**

**Practical 1**

Auteur : F. Hurkmans

Datum : 29 Augustus 2014

Versie : 1.2

**Practical 1: Meetserie**

Onder een meetserie wordt in deze opgave verstaan:

een serie meetgegevens (int waarden), eventueel voorzien van een naam.

Een voorbeeld:

Bij afslag 29 van de autosnelweg A2 heeft men een detectielus geplaatst om te meten hoeveel voertuigen passeren en hoe zwaar die voertuigen zijn.

Die detectielus bepaalt hoe zwaar het passerende voertuig is (afgerond op gehele tonnen) en plaatst dat in een meetserie.

Een concrete meetserie is dan bijvoorbeeld:

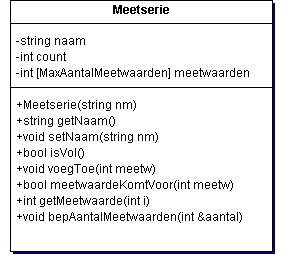
|  |  |
| --- | --- |
| Naam | "afslag 29 van de A2" |
| aantal metingen | 10 |
| meetwaarden | 1, 1, 6, 1, 12, 1, 1, 2, 20, 2 |

Er zijn dus 10 metingen verricht: een 1-tonner, een 1-tonner, een 6-tonner, een 1-tonner, een 12-tonner, een 1-tonner, een 1-tonner, een 20-tonner en een 2-tonner.

Er is een klasse Meetserie ontworpen om de gegevens van een meetserie te kunnen opslaan. In Figuur 1 is deze klasse weergegeven.

In C++ slaan we een klasse, zoals Meetserie, op in 2 aparte bestanden.

Het eerste bestand, de headerfile (.h), bevat de definitie van de klasse en het tweede bestand, de codefile (.cpp), bevat de implementatie van de methoden die bij die klasse horen.



Figuur 1 : UML klasdiagram Meetserie

De headerfile, Meetserie.h, is hieronder weergegeven.

#ifndef MeetserieH

#define MeetserieH

const int MaxAantalMeetwaarden = 10;

class Meetserie

{

private:

string naam; // de naam van <Meetserie>

int count; // het aantal meetwaarden van Meetserie

int meetwaarden[MaxAantalMeetwaarden]; // meetwaarden[0..count-1]

// zijn de meetwaarden

public:

Meetserie(string nm);

/\* pre : -

post: Meetserie bevat geen meetwaarden en de

naam van Meetserie is nm

\*/

string getNaam();

/\* pre : -

post: return de naam van Meetserie

\*/

void setNaam(string nm);

/\* pre : -

post: de naam van Meetserie is nm

\*/

bool isVol();

/\* pre : -

post: Als het aantal meetwaarden van Meetserie is maximaal

Dan true

Anders false

\*/

void voegToe(int meetw);

/\* pre : het aantal meetwaarden van Meetserie is niet maximaal

post: meetw is aan de meetwaarden van Meetserie toegevoegd

\*/

bool meetwaardeKomtVoor(int meetw);

/\* pre : -

post: Als meetw voorkomt onder de meetwaarden van Meetserie

Dan return true

Anders return false

\*/

int getMeetwaarde(int i);

/\* pre : 0<=i<aantal meetwaarden

post: return i-de meetwaarde

\*/

void bepAantalMeetwaarden(int &aantal);

/\* pre : -

post : variabele “aantal” is gelijk aan het

aantal meetwaarden van <Meetserie>

\*/

};

#endif

Maak een leeg project aan in code::blocks en voeg hieraan de code file main.cpp toe voor het handmatig testen van de klas en een headerfile en een code file toe voor de klasse Meetserie. (In de headerfile van Meetserie moeten nog een include en een using geplaatst worden om string te kunnen te gebruiken.)

Realiseer de gevraagde functionaliteit en test deze middels het aanroepen van de functionaliteit in een menu gemaakt in de code file Opdracht1.cpp.

(1) Tonen alle waarden

(2) Toevoegen van een meetwaarde

(3) Testen of er nog meeetwaarden geplaatst kunnen worden

(4) Tonen hoevaak een bepaalde meetwaarde voorkomt

(5) Tonen van de naam van de meetserie

(6) Aanpassen van de naam van de meetserie

(7) Testen of een meetwaarde voorkomt

(9) STOPPEN

Maak 2 build targets aan in code::blocks, target “debug” en target “test” om zowel handmatig als automatisch te kunnen testen. Gebruik google-test voor de automatische testen en maak een MeetserieTest klasse aan voor de unit test van klasse Meetserie. Kijk voor inspiratie evt. bij de code van PRC31 voorbeeld “Spaarpot”.

**Bonus:**

Breid klasse Meetserie uit met een save en een load method om een Meetserie op te kunnen slaan in een file “meetserie.txt”.

Breid het menu uit met twee keuzes om de nieuwe save en load functionaliteit te testen voor m1.