

Pointer-practicum

13/14 november 2025

1 Uitleg

Het doel van het practicum is het begrijpen van *pointers*. Kopieer daartoe de file `oefpoint.cc` naar je eigen account. Dit programma compileert al, maar doet nog niet zo veel. Maak de functies I, II en III — in die volgorde.

De bedoeling is dat er een lijst wordt opgebouwd (zie pagina 2) waarbij in ieder **vakje** een integer wordt opgeslagen (in het eerste **vakje** uit de lijst: 34), en verder een pointer naar een andere integer (bij het eerste **vakje** 23) en een pointer naar het volgende **vakje**.

Je bent natuurlijk naar het “pointer-college” van donderdag 13 november geweest. en de bijbehorende sheets: www.liacs.leidenuniv.nl/~visjk/pm/bestanden/pointer.pdf.

Als onderstaande allemaal echt klaar is, “doe” dan ter voorbereiding op de vierde programmeeropgave www.liacs.leidenuniv.nl/~visjk/pm/bestanden/dubbel.cc.

2 Code

Het programma `oefpoint.cc` is te vinden via

www.liacs.leidenuniv.nl/~visjk/pm/bestanden/oefpoint.cc

(zie onder). Een uitgewerkte versie staat hier:

www.liacs.leidenuniv.nl/~visjk/pm/bestanden/pointer.cc

— niet spieken!

```
1
2 //
3 // Programmeermethoden 2025
4 //
5 // http://www.liacs.leidenuniv.nl/~visjk/pm/bestanden/oefpoint.cc
6 // C++-programma om te oefenen in het gebruik van pointers.
7 // Er moet eerst een enkelverbonden pointerlijst gefabriceerd worden,
8 // waarbij de "vakjes" die met elkaar verbonden worden bestaan uit
9 //   1. Een int(eger)
10 //   2. Een pointer naar een int(eger) (of nullptr)
11 //   3. Een pointer naar het volgende vakje (of nullptr)
12 // Schrijf de functies Afdrukken, Toevoegen en Verwijderen
13 // --- in deze volgorde.
14 // Er is ook een uitwerking van dit programma beschikbaar:
15 //   http://www.liacs.leidenuniv.nl/~visjk/pm/bestanden/pointer.cc
16 //
17 // Doe hierna
18 //   http://www.liacs.leidenuniv.nl/~visjk/bestanden/pm/dubbel.cc
19 //
20 // Compiler: GNU g++
21 // Datum: 10 november 2025
22 // Auteur: Jonathan Vis, Informatica Universiteit Leiden
```

```

23 //          e-mail pm@liacs.leidenuniv.nl
24 //
25
26 #include <iostream>
27 using namespace std;
28
29 // het soort vakje waar het om draait:
30 class vakje {          // een struct mag ook
31     public:
32         int info;
33         int* andere;
34         vakje* volgende;
35 }; //vakje
36
37 //
38 //   +-----+          +-----+-----+          +-----+-----+
39 //   |  --+-----> | 34 | | |  --+-----> | 56 | | | null|ptr
40 //   +-----+          +-----+-----+          +-----+-----+
41 //   ingang              |                      |
42 //                       V                      V
43 //                       +-----+          +-----+
44 //                       | 23 |              | 18 |
45 //                       +-----+          +-----+
46 //
47 // ingang wijst dus een vakje uit een lijst vakjes aan
48 // waarbij ieder vakje informatie bevat (het info-veld),
49 // een pointer naar een int(eger) (het andere-veld),
50 // en een pointer naar het volgende vakje; om zulke
51 // structuurtjes te maken is doorgaans in totaal steeds
52 // twee maal new nodig, voor het voorbeeld hierboven in totaal vier
53 // Een voorbeeld: het getal 34 (in het eerste vakje)
54 // wordt afgedrukt via:
55 //   cout << ingang->info << endl;
56 // En het getal 18 (toegankelijk door het tweede vakje) via:
57 //   cout << *(ingang->volgende->andere) << endl;
58 // En een pointer hulp (vakje* hulp) gaat naar het vakje
59 // met onder andere 56 erin wijzen via:
60 //   hulp = ingang->volgende;
61 //
62
63 // *****
64
65 void Afdrukken (vakje* ingang) {
66 // druk lijst met ingang als ingang af
67     cout << "Lijst afdrukken ..." << endl;
68
69     //
70     //   XXXXXX
71     //   XX
72     //   XX
73     //   XXXXXX
74     //
75
76 } //Afdrukken
77
78 void Toevoegen (vakje* & ingang, int een, int twee) {

```

```

79 // voeg vakje met getallen een en twee erin vooraan lijst ingang toe
80 // preciezer: nieuw vakje met getal een en een POINTER naar een
81 // nieuwe int met getal twee erin
82 // mooier: als bijvoorbeeld twee = -1, dan p->andere nullptr maken
83 cout << "Voeg een nieuw vakje toe ..." << endl;
84
85 //
86 //   XXXXXX XXXXXX
87 //   XX     XX
88 //   XX     XX
89 //   XXXXXX XXXXXX
90 //
91
92 }//Toevoegen
93
94 void Verwijderen (vakje*& ingang) {
95 // gooi eerste vakje van lijst ingang weg als ingang niet nullptr is
96 cout << "Verwijder een vakje ..." << endl;
97
98 //
99 //   XXXXXX XXXXXX XXXXXX
100 //   XX     XX     XX
101 //   XX     XX     XX
102 //   XXXXXX XXXXXX XXXXXX
103 //
104
105 }//Verwijderen
106
107 void LeesInGetallen (int & een, int & twee) {
108 // lees integers een en twee in
109 cout << "Eerste getal svp .. ";
110 cin >> een;
111 cout << "Tweede getal svp .. ";
112 cin >> twee;
113 }//LeesInGetallen
114
115 // *****
116
117 int main ( ) {
118
119     vakje* ingang = nullptr; // ingang van de op te bouwen lijst (of NULL)
120     char keuze;           // wat wil de gebruiker?
121     int een, twee;      // toe te voegen getallen
122
123     do {
124         cout << "Kies uit: [s]toppen, [t]oevoegen, " << endl
125             << "                [a]fdrukken, [v]erwijderen" << endl
126             << "Uw keuze: ";
127         cin >> keuze;
128         switch (keuze) {
129             case 's': case 'S':
130                 cout << "Dat was het dan ..." << endl;
131                 break;
132             case 't': case 'T':
133                 LeesInGetallen (een,twee);
134                 Toevoegen (ingang,een,twee);

```

```

135     break;
136     case 'a': case 'A':
137         Afdrukken (ingang);
138         break;
139     case 'v': case 'V':
140         Verwijderen (ingang);
141         break;
142     default:
143         cout << "Niet toegestane menukeuze ..." << endl;
144     }//switch
145 }//do
146 while ( ! ( keuze == 's' ) && ! ( keuze == 'S' ) );
147
148 // while ( ingang != nullptr ) // netjes opruimen
149 //     Verwijderen (ingang);
150
151 return 0;
152
153 }//main

```