

Jun 18 2007 18:50

Klasse.h

Page 1

```

1 //=====
2 //      * Letsch Informatik *      www.LetsInfo.ch      CH-8636 Wald
3 //      Beratung, Ausbildung und Realisation in Software-Engineering
4 //=====
5 // Project   : C++-Kurs @ Sultex Limited
6 // Modul     : C++
7 // Title     : Übung 2 "Klasse": Loesung
8 // Author    : Thomas Letsch
9 // Tab-Width : 2
10 /*///=====
11
12 * Description:
13 Klasse implementieren aufgrund applikatorischer Vorgaben.
14 Mit Strings als char-Array arbeiten.
15 Bearbeitung von Arrays aus Objekten.
16 Allozierung auf Stack und Heap vergleichen.
17 Performanceuntersuchung mit inline-Methoden (-> Kompiler-Option: -O2)
18
19 * History   :
20 02.01.04: Initial Version.
21 29.10.05: File-Split wegen inline.
22 * Version   : $Revision: 1.7 $ $Date: 2007/06/18 18:49:49 $
23 /*///=====
24 //      1      2      3      4      5      6      7      8
25 //34567890123456789012345678901234567890123456789012345678901234567890
26 //=====
27 #ifndef KLASSE_H
28 #define KLASSE_H 1
29
30 #include <string>
31 #include <iostream>
32
33 #include "StatisticStopWatch.h"
34
35 using namespace std;
36
37
38 // Klassen-Definitionen:
39
40 // Person:
41 // Eine Person hat einen Namen (20 Char's) und eine Nummer.
42 class Person {
43
44     enum {NAME_LEN = 20};
45
46     public:
47
48         Person();
49         Person(int pNr, const char* pName);
50         int   getNr();
51         int   getNrInline();
52         const char* getName();
53         void  setNr(int pNr);
54         void  setName(const char* pName);
55
56     private:
57         int   mNr;
58         char  mName[NAME_LEN+1];
59
60 };
61

```

Jun 18 2007 18:50

Klasse.h

Page 2

```

62
63 // Funktions-Prototypen:
64
65 // Ausgabe eines Arrays von Personen-Objekten auf die Konsole.
66 // pPers: Pointer auf erstes Personen-Objekt im Array.
67 // pLen: Laenge des Personen-Arrays.
68 void printPersArr(Person* pPers, int pLen);
69
70
71 // Inline-Methoden:
72
73 inline int Person::getNrInline() {
74     return mNr;
75 }
76
77 #endif /*KLASSE_H*/

```

Oct 3 2006 18:21

Klasse.cpp

Page 1

```

1 #include "Klasse.h"
2
3 // Methoden-Implementationen:
4 Person::Person() {
5     mName[0] = '\0'; // Null-Byte als erstes Byte im Array.
6     mNr = -1;
7 }
8
9
10 Person::Person(int pNr, const char* pName) {
11     mNr = pNr;
12     setName(pName);
13 }
14
15
16 int Person::getNr() {
17     return mNr;
18 }
19
20
21 const char* Person::getName() {
22     return mName;
23 }
24
25
26 void Person::setNr(int pNr) {
27     mNr = pNr;
28 }
29
30
31 void Person::setName(const char* pName) {
32     strcpy(mName, pName);
33 }
34
35
36
37 // Funktions-Implementationen:
38
39 void printPersArr(Person* pPers, int pLen) {
40     for (int i = 0; i < pLen; i++) {
41         cout << pPers->getNr() << ": " << pPers->getName() << endl;
42         pPers++; // pPers soll auf 'naechste' Person zeigen (-> Zeiger-Arithmetik)
43     }
44 }
45
46

```

Oct 3 2006 18:21

KlasseTest.cpp

Page 1

```

1 #include "Klasse.h"
2
3 int main() {
4
5     const int MAX_ARR = 5;
6
7     Person persArr[MAX_ARR];
8     for (int i = 0; i < MAX_ARR; i++) {
9         cout << persArr[i].getNr() << ": " << persArr[i].getName() << endl;
10    }
11    persArr[0].setName("Miller");
12    persArr[0].setNr(1);
13    persArr[1].setName("Bond");
14    persArr[1].setNr(007);
15    for (int i = 0; i < MAX_ARR; i++) {
16        cout << persArr[i].getNr() << ": " << persArr[i].getName() << endl;
17    }
18
19    // Ausgabe des Personen-Arrays wie oben mit for-Schleifen, jetzt aber mit
20    // Funktion 'printPersArr()':
21    printPersArr(persArr, MAX_ARR);
22
23
24
25    // Untersuchung wieviel Zeit Objekt-Allozierungen auf Stack und Heap benoetigen:
26
27    StatisticStopWatch stopWatch;
28    const int MAX_LOOP = 5000000;
29
30
31    // Allozierung auf dem Stack:
32    stopWatch.reset();
33    stopWatch.start();
34    for (int i = 0; i < MAX_LOOP; i++) {
35        Person pers(i, "John");
36    }
37    stopWatch.stop();
38    stopWatch.printTime("Stack");
39
40
41    // Allozierung auf dem Heap:
42    stopWatch.reset();
43    stopWatch.start();
44    for (int i = 0; i < MAX_LOOP; i++) {
45        Person* pers = new Person(i, "John");
46    }
47    stopWatch.stop();
48    stopWatch.printTime("Heap: new()");
49

```

Oct 3 2006 18:21

KlasseTest.cpp

Page 2

```

50 |
51 | // Tests mit 'inline-Methoden':
52 | int max_loop = 100 * MAX_LOOP;
53 | stopWatch.reset();
54 | stopWatch.start();
55 | Person pers;
56 | for (int i = 0; i < max_loop; i++) {
57 |     pers.getNr();
58 | }
59 | stopWatch.stop();
60 | stopWatch.printTime("getNr()");
61 | stopWatch.reset();
62 |
63 | stopWatch.reset();
64 | stopWatch.start();
65 | for (int i = 0; i < max_loop; i++) {
66 |     pers.getNrInline();
67 | }
68 | stopWatch.stop();
69 | stopWatch.printTime("getNrInline()");
70 | stopWatch.reset();
71 |
72 | }
73 |
74 |
75 |
76 |
77 |
78 | /* Session-Log (Intel Pentium M / 1700 MHz; g++ 3.4.4):
79 |
80 | $ make clean
81 | rm -f Klasse.o KlasseTest.o StatisticStopWatch.o appl.exe
82 |
83 | $ make all
84 | g++ -g -O2 -c Klasse.cpp
85 | g++ -g -O2 -c KlasseTest.cpp
86 | g++ -g -O2 -c StatisticStopWatch.cpp
87 | g++ -g -O2 -o appl.exe Klasse.o KlasseTest.o StatisticStopWatch.o
88 |
89 | $ appl
90 | -1:
91 | -1:
92 | -1:
93 | -1:
94 | -1:
95 | 1: Miller
96 | 7: Bond
97 | -1:
98 | -1:
99 | -1:
100 | 1: Miller
101 | 7: Bond
102 | -1:
103 | -1:
104 | -1:
105 | Stack                =          218 ms
106 | Heap: new()           =         7657 ms
107 | getNr()               =         4140 ms
108 | getNrInline()        =         1031 ms
109 |
110 | $
111 |
112 | */
113 |

```

Jun 18 2007 18:53

Makefile

Page 1

```

1 |
2 | # Makefile für "Klasse"                                18.06.2007
3 |
4 | BIN      = appl.exe
5 | OBJS     = Klasse.o KlasseTest.o StatisticStopWatch.o
6 |
7 | CC       = g++
8 | CFLAGS   = -g -O2
9 |
10 | all: $(BIN)
11 |
12 | $(BIN): $(OBJS)
13 |     $(CC) $(CFLAGS) -o $(BIN) $(OBJS)
14 |
15 | clean:
16 |     rm -f $(OBJS) $(BIN)
17 |
18 | %.o: %.cpp %.h
19 |     $(CC) $(CFLAGS) -c $<
20 |
21 | %.o: %.cpp
22 |     $(CC) $(CFLAGS) -c $<

```