ETH Zürich

Informatik I
Übungsstunde 05

Johannes Odenthal

cs.jodenthal.com

Hausaufgabenreview · Exercise 04

Factorial · Ausschnitt

Don’t

```
if (n > 0) {
  while (n > 0) {
    result *= n;
    --n;
  }
}
```

Do

```
while (n > 0) {
  result *= n;
  --n;
}
```

Repeated Addition · Bedingung

Don’t

```
n == 1
```

Do

```
n != 0
```

Powers of Two

Don’t

Nur einige Werte testen

Do

Basisfälle und Induktionsschritt begründen

32-Bit-int: $0 \leq n \leq 30$

Hausaufgabenfragen

Welche Fragen zu den Schleifenaufgaben sind offen?

Rounding Numbers

CodeExpert

Debugging with Assert

CodeExpert

Functions

Pre-Condition und Post-Condition

```
1  double f(double i, double j,
2           double k) {
3    if (i > j) {
4      if (i > k) { return i; }
5      else { return k; }
6    } else {
7      if (j > k) { return j; }
8      else { return k; }
9    }
10 }
```

Was macht die Funktion f?

Post-Condition: liefert das Maximum von i, j und k
Für gewöhnliche endliche Zahlen keine zusätzliche Pre-Condition nötig

Ein Vertrag für eine Summe

```
1 double g(int i, int j) {
2   double r = 0.0;
3   for (int k = i; k <= j; ++k) {
4     r += 1.0 / k;
5   }
6   return r;
7 }
```

Was wird berechnet? Wann ist der Aufruf sinnvoll?

Post-Condition: Summe von 1/i bis 1/j
Pre-Condition: i ≤ j, j < INT_MAX und 0 liegt nicht im Bereich

Funktionsaufrufe nachvollziehen

```cpp
int f(int i) { return i * i; }

int g(int i) {
  return i * f(i) * f(f(i));
}

void h(int i) {
  std::cout << g(i) << "\n";
}

int main() {
  int i;
  std::cin >> i;
  h(i);
}
```

Was gibt dieses Programm für i = 2 aus?

$f(i) = i^2 \cdot f(f(i)) = i^4$

$g(i) = i^7 \cdot$ Ausgabe für i = 2: 128

Functions · Drei Fehler

```cpp
// Absichtlich fehlerhaft
double f(double x) {
  return g(2.0 * x);
}

bool g(double x) {
  return x % 2.0 == 0;
}

void h() {
  std::cout << result;
}

int main() {
  double result = f(3.0);
  h();
}
```

Welche drei Stellen verhindern das Kompilieren?

g ist beim Aufruf noch nicht deklariert
Modulo verlangt ganzzahlige Operanden
result ist in h nicht sichtbar

Math functions

CodeExpert

Fragen?

Headers und Namespaces

CPP Headers & Namespaces

CodeExpert

Fragen?

Stepwise Refinement

Perfect Numbers

CodeExpert

Fragen?

Optional · Nachschlagen

Optional · Number of Divisors

```c
// PRE: 1 <= n < INT_MAX
int number_of_divisors(int n) {
  // Absichtlich unvollstaendig
}
```

Wie viele Teiler haben 1, 6 und 12?

1 ⟶ 1 · 6 ⟶ 4 · 12 ⟶ 6

Optional · Teiler zählen

```
1  // PRE: 1 <= n < INT_MAX
2  int number_of_divisors(int n) {
3    int count = 0;
4    for (int d = 1; d <= n; ++d) {
5      if (n % d == 0) {
6        ++count;
7      }
8    }
9    return count;
10 }
```

Alle d von 1 bis n werden geprüft
Nur Teiler erhöhen count

Nachschlagen · Standardbibliothek

```
#include <cmath>
std::pow(2.5, 2)     // 6.25
std::sqrt(14.0625)   // 3.75
std::abs(-3.5)       // 3.5

#include <algorithm>
std::min(2.0, 1.0)   // 1.0
std::max(2, 1)       // 2
```

Bei min und max müssen die beiden Argumenttypen zusammenpassen

Optional · Exam 2022-08 MAVT

auto übernimmt den Typ des Initialisierungsausdrucks

```
1 int a = 5;
2 int b = 1;
3 auto c = (9 * a + b) % a;
```

```
1 int a = 5;
2 double b = 1;
3 auto c = (9.0 * a + b) / a;
```

Welchen Typ und Wert hat c jeweils?

Links: int · 46 % 5 = 1
Rechts: double · 46.0 / 5.0 = 9.2

Optional · Loop Termination

```
1 int sum = 17;
2 int i = 1;
3 do {
4   i += sum;
5   sum = sum / 2;
6 } while (i > sum && sum >= 0);
7 std::cout << sum;
```

Nach dem ersten Rumpf: i = 18, sum = 8
Danach: (i, sum) = (26, 4), (30, 2), (32, 1), (33, 0)
Ab dann bleiben i = 33 und sum = 0
Die Bedingung bleibt true · keine Ausgabe