

Übung Softwarequalität TINF21

Übung Nr.6

Hinweise:

Bei Multiple Choice Aufgaben sind mehrere Antworten möglich.

Softwarefehler

Aufgabe 1:

Worauf ist bei Verwendung von Gleitkommazahlen in einem Programm zu achten?

- a) ☐ Gleitkommazahlen laufen schnell über, so daß eine Skalierung erforderlich ist.
- b) ☐ Die Darstellung im Binärformat des Computers erlaubt keinen exakten Vergleich.
- c) ☐ Nicht jede Zahl ist endlich im Binärformat darstellbar, dadurch ergeben sich Ungenauigkeiten.
- d) ☐ Gleitkommazahlen erlauben keine negativen Werte.

Konstruktive Qualitätsmaßnahmen

Aufgabe 2:

Das nachfolgende C-Programm implementiert einen Frostwarner. Zeigen Sie den Fehler und ändern Sie das Programm, so daß dieser Fehler vermieden wird.

```
/* gibt 0 zurück, wenn kein Frost vorherrscht, ansonsten eine 1 */
unsigned int isFrost( int temperature )
{
    /* gets the temperature and returns frost condition */
    /* when temperature is below 32degFahrenheit */
    if( temperature <= 32 )
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

void main( void )
{
    int temperatur = 32;    /* Temperatur in GradCelsius */

    if( isFrost( temperatur ) == 1 )
    {
        printf("Wir haben Frost!");
    }
}
```

Aufgabe 3:

Folgender C-Programmauszug enthält verschiedenartige Fehler bzw. Problemstellen. Identifizieren Sie die Problemstellen und beschreiben Sie kurz mögliche Lösungsmöglichkeiten, um den Fehler zukünftig zu **vermeiden**.

```
1  double sum_waypoint( int* waypoint_meter, int size, Point coor )
2  {
3      double correction_fact=1.0;
4      double waysum;
5      int it;
6      /* holt einen Korrekturfaktor auf den Wegpunkt
7       * für die aktuelle Koordinate anhand einer Karte
8       * Der Korrekturfaktor ist gültig für metrische Einheiten
9
10     correction_fact = get_position_correction( coor ); /* benutze API Funktion */
11     for(it=0; it < size; it++);
12         waysum += waypoint_meter[it]*correction_fact;
13     return waysum;
14 }
15
16 int main( void )
17 {
18     int waypoints[] = { 05, 7, 15, 40}; /* Wegpunkte in Fuß */
19     double sum;
20
21     /* berechne die Summe der korrigierten Wegpunkte */
22     sum = sum_waypoints( waypoints, sizeof(waypoints), CONST_EUROPE);
23
24     printf("Wegpunktsumme %f\n", sum );
25
26
27     return(0);
28 }
29
```