

◀ Info2 (Ahaus) – Sommersemester 17

Aufgabenblatt 7: Klassen, Templates und Exceptions

Kapitel 1: Praktikumsaufgaben

7.1.1: Vervollständigung einer Klasse ✓

Sie haben die folgende Klasse ohne besondere Bedeutung gegeben:

```
class speicher
{
public:
    char c;
    speicher( char init ) { c = init; }
};
```

Von dieser Klasse wollen Sie eine weitere Klasse ableiten

```
class meinspeicher : public speicher
{
private:
    speicher* pointer;
};
```

Ergänzen Sie meinspeicher mit Konstruktor, Copy-Konstruktor und Destruktor so, dass alle Member korrekt initialisiert, kopiert und beseitigt werden. Die beiden zur Konstruktion benötigten char-Parameter sollen dabei als Parameter an den Konstruktor von meinspeicher übergeben werden und der Speicher für die Verwendung über den Zeiger pointer soll dort im Konstruktor allokiert werden.

Feedback

Du hast am 30.06.17, 14:05 folgendes Feedback zu dieser Aufgabe abgegeben:

bearbeitet: komplett

Schwierigkeit: 5/10

Spaß: 2/10

Zeit: 10min

Text: komisch

7.1.2: Keylogger

Sie wollen die Tastendrücke in einem Programm protokollieren. Erstellen Sie dazu eine Klasse keylog, die folgendes leistet:

- Bei der Konstruktion der Klasse kann festgelegt werden, wie viele Tastendrücke maximal gespeichert werden sollen. Der Default dafür ist 100.
- Entsprechend der Kapazität können Tastendrücke über eine Memberfunktion gespeichert werden.
- Ein Tastendruck ist über einen Zahlenwert (int) gegeben.
- Zusätzlich wird zu jedem Tastendruck ein Zeitstempel gespeichert. Wir nehmen dazu an, dass ein Zeitstempel ein int-Wert ist, den man über die Funktion `gettimestamp()` holen kann.
- Wird die Kapazität überschritten, so wird der am längsten zurückliegende Tastendruck verworfen.

- Die Klasse hat eine Methode, die die Zahlenwerte der Tastendrucke und deren Zeitstempel auf dem Bildschirm ausgibt.

7.1.3: Ausnahmebehandlung

Sie wollen eine Funktion erstellen, die zwei Integer-Parameter x und y übergeben bekommt. Dabei muss der Parameter x stets ungleich 0 und kleiner als y sein.

Die sich dadurch ergebenden Ausnahmefälle wollen sie durch Exception-Handling behandeln. Das rufende Programm soll dabei über eine Klasse die Information erhalten, welche der beiden Ausnahmen eingetreten oder ob sogar beide Ausnahmen eingetreten sind.

Erstellen Sie die Funktion und zusätzlich ein Programm, dass diese Funktion für alle Parameterkombinationen ($-10 < x, y < 10$) ruft und dabei die auftretenden Exceptions mit den zugehörigen Parameterwerten auf dem Bildschirm protokolliert.

7.1.4: Ausnahmefallbehandlung für die generische Stack-Klasse

Erweitern Sie die generische Klasse `stack` aus der Vorlesung um eine geeignete Ausnahmefallbehandlung. Erstellen Sie für die Ausnahmen geeignete Klassen.

```
template<typename T, int SIZE>
class stack
{
    private:
        T stck[SIZE];
        int top;

    public:
        stack() { top = 0; }
        int push( T element );
        T pop();
        int isEmpty(){ return top == 0; }
};

template<typename T, int SIZE>
int stack<T, SIZE>::push( T element ) {
    if( top < SIZE )
    {
        stck[top] = element;
        top++;
        return 1;
    }
    return 0;
}

template<typename T, int SIZE>
T stack<T, SIZE>::pop()
{
    if( top > 0 )
        top--;
    return stck[top];
}
```

Erstellen Sie ein Programm, dass das Template verwendet und testen Sie alle entstehenden Ausnahmefälle und

deren Behandlung anhand des Tests.