

Übungsstunde

Woche 14

Adel Gavranović

adel.gavranovic@inf.ethz.ch

Übersicht

Heutige Themen

Intro

Inheritance und Polymorphism

Prüfungsvorbereitung

Fragen

Simplifying `if()`-chains

Outro

Links

▶ [polybox zum Material für die Übungsstunden](#)

▶ [Mail an Assistenten](#)

Sad Deli Noises

Vorerst letzte Übungsstunde zusammen :’(

Follow-up aus vorheriger Übungsstunde

- Was passiert wenn man versucht `delete [] ++ptr; ?`
 - Undefined Behavior!

Kommentare zu [code] expert

- Ich werde mir Mühe geben, die Korrekturen asap zu erledigen
- Entschuldigung für den Verzug

Fragen zu [code]expert eurerseits?

Lernziele

- Subclasses verstehen
- Unterschied zwischen `virtual` und non-`virtual` verstehen
- Class-Hierarchien implementieren können

Linked List Recap?

New Node oder no new Node?

Eine Funktion soll eine neue Node kreieren und einen Pointer darauf returnen. In der Funktion sehen wir folgenden Code

```
1 Node new_node(value);  
2 return &new_node;
```

Frage

Das Programm verhält sich nicht immer so, wie wir es möchten.
Wieso?

Antwort

Die Node wurde *nicht dynamisch* alloziert, daher wird sie am Ende des Scopes auch wieder dealloziert. Die Adresse, auf die der Pointer zeigt existiert zwar noch, aber der Inhalt könnte bereits ein anderer sein. Verhalten wieder undefined.

New Node or no new Node?

Frage

Wie umschreiben, damit sie funktioniert wie gewollt?

```
1 // allocate dynamically!  
2 Node* new_node = new Node(value);  
3 return new_node;
```

Nicht vergessen

Dynamisch allozierter Speicher (also mit `new` alloziert) wird *nicht* automatisch wieder am Ende des Scopes dealloziert, aber *normal/statisch* allozierter Speicher schon.

Fragen or Comments re: Exercises?

Inheritance and Polymorphism

nicht prürel, *also...*
nur ein kleiner konzeptioneller Überblick

Inheritance and Polymorphism

Fragen/Unklarheiten?

”iSt DaS PRÜfUnGsReLeVaNt?”

- Alles, was nicht explizit als *nicht prüfungsrelevant* gekennzeichnet wurde, *ist* prüfungsrelevant
- Fertigt ein gigantisches PDF von allen Vorlesungshandouts an und `ctrl+F` durch das Dokument, wenn ihr euch bei einem Thema nicht mehr sicher seid
- Sachen, die auf den Vorlesungshandouts sind, aber über die nicht viel in den Vorlesungen geredet wurde, könnten noch immer an der Prüfung vorkommen (bspw. $010 = 8_{10}$)

An der Prüfung

- Warnings \neq Errors
- Punkte werden proportional zu bestandenen Testcases vergeben
- Physische Keyboards: nur CH/DE und US/INT (in den Settings glaube ich noch CH/FR)
- Keine Chance mit Hardcoding

Punkte zur Prüfung

- Ihr werdet einen Skill aufbauen und üben müssen. Einfach nur "wissen" wie es geht *reicht nicht*. **Üben, Üben, Üben.**
- Übt mit dem Tastaturlayout, dass ihr dann auch an der Prüfung verwenden werdet (CH-DE/US)
- @MacUsers: Ihr werdet eine Art "Cheat Sheet" an der Prüfung ausgeteilt bekommen, aber davor schon mit einer non-Apple Tastatur zu üben vielleicht sinnvoll
- Benutzt nichts an der Prüfung, dass nicht in der Vorlesungen drankam insb. libraries und data structures (stacks, heaps, queues, cmath, ...). Könnte euch unnötigerweise kostbare Punkte kosten
- Kein Bedarf für gute Dokumentation an der Prüfung (ausser es ist explizit danach gefragt). Ähnliches gilt für `const` (aber `const` Memberfunktionen müsst ihr beherrschen!)

Punkte zur Prüfung

- Bin auch während der Lernphase anmailbar. Wenn es etwas zu kompliziertes ist, können wir auch kurz zoomen
- Übt mit den Mock-exams und den alten Programmierprüfungsaufgaben
- Vergesst nicht, dass ihr auch andere Prüfungen habt und verbringt nicht eure gesamte Zeit mit Informatik
- *IMO*: Speedruns üben hilft dabei, dann an der Prüfung ruhiger zu sein
- Bringt Notizpapier mit; es macht einige Fragen lächerlich einfach
- *Excited not panicky*
- Übertreibt es nicht mit Koffein vor der Prüfung (oder Alkohol danach)

Q&A

Q&A

Q&A

Q&A

Q&A

Q&A

Q&A

Q&A

Q&A

What's an if()-chain?

```
1  if(grid != nullptr){
2      if(grid -> is_filled(row, col)){
3          if(col == 8){
4              if(fillValidNumber(grid, row + 1, 0)){
5                  return true;
6              }
7          } else {
8              if(fillValidNumber(grid, row, col + 1)){
9                  return true;
10             }
11         }
12     }
```

Task

Simplify this mess

Simplifying...

```
1     if(grid != nullptr){
2         if(grid -> is_filled(row, col)){
3             if(col == 8){
4                 return fillValidNumber(grid, row + 1, 0);
5             } else {
6                 return fillValidNumber(grid, row, col + 1);
7             }
8         }
9     }
```

Task

Simplify this even further

Simplifying further...

```
1     if(grid != nullptr && grid -> is_filled(row, col)){
2         if(col == 8){
3             return fillValidNumber(grid, row + 1, 0);
4         } else {
5             return fillValidNumber(grid, row, col + 1);
6         }
7     }
```

Trick

Two `if()` can sometimes be put together into a one boolean expression with `&&`

Remember

`&&` *shortcircuits*, so it won't check the second one if the first one returns true

Fragen/Unklarheiten?

VIELEN DANK!

Frohe Festtage, gutes Neues und
viel Erfolg bei den Prüfungen