

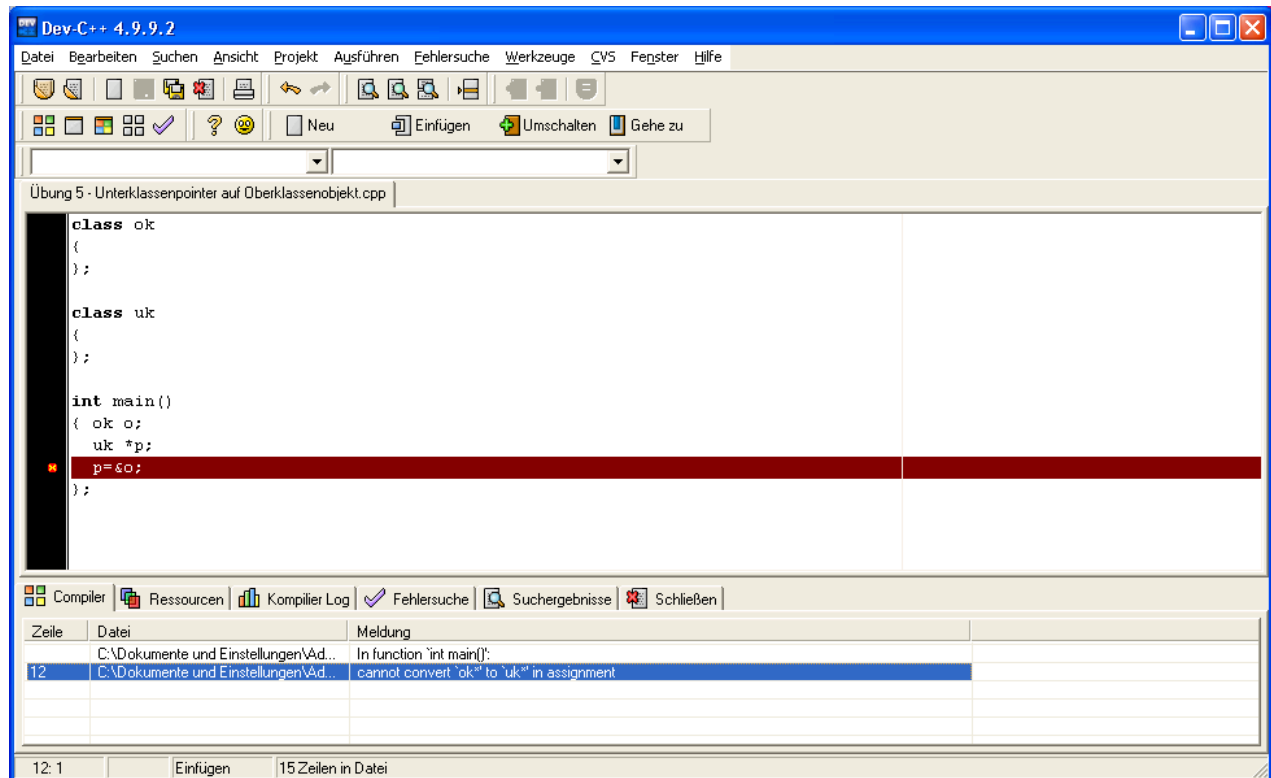
Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Objektorientierte Programmierung	SS 2019
Übung 5	Termin 25.4.19

Klassen & Vererbung

1. Aufgabe: Polymorphismus

Polymorphismus bedeutet „Vielgestaltigkeit“. Im Sinn der Objektorientierten Programmierung kann eine Pointervariable vom Typ Pointer auf eine Klasse K auch auf Objekte von Unterklassen von K zeigen. Überlegen Sie, warum das umgekehrt nicht gilt, d.h. warum ein Unterklassenpointer nicht auf ein Objekt der Oberklasse zeigen kann.
Probieren Sie aus, was passiert, wenn Sie das in einem Programm versuchen.

An einem Pointer auf ein Unterklassenobjekt kann eine Methode aufgerufen werden, die in der Oberklasse nicht existiert und dort u.U. keinen Sinn macht. Des weiteren könnte über einen Unterklassenpointer auf ein öffentliches Attribut zugegriffen werden, das in der Oberklasse nicht vorhanden ist. Wenn ein Unterklassenpointer also auch auf ein Oberlassenobjekt zeigen könnte, würde auf ein im Oberlassenobjekt nicht vorhandenes Element (Methode oder Attribut) zugegriffen. Siehe auch Compiler-Output unten.



2. Aufgabe: Redefinition und Dynamisches Binden: Datumsausgabe

- a.) Schreiben Sie eine Klasse Datum, die Jahreszahl, Monat und Tag in geeigneter Form in Attributen speichert. Die Attribute sollen Unterklassen zur Verfügung stehen, aber nicht öffentlich sein. Die Datumsklasse soll eine Methode besitzen, die ein Datum von Tastatur einliest und die Attribute setzt und eine zweite Methode, die das Datum (im Beispiel der 10.5.2010) in folgender Form ausgibt:

Jahr: 2010
 Monat: Mai
 Tag: 10

Vereinbaren Sie ein Objekt der Klasse Datum und einen Pointer auf ein Objekt der Klasse Datum und probieren Sie die Datumsausgabe aus.

- b.) Schreiben Sie zwei Unterklassen D_Datum (wie „Deutsches Datum“) und U_Datum (wie „US-Datum“) der Klasse Datum, in denen Sie die Methode zur Ausgabe des Datum jeweils redefinieren (d.h. die Ausgabefunktionen der Unterklassen haben den gleichen Namen und die gleiche Signatur). Dabei soll die Ausgabe des Datums in der Klasse D_Datum in der Form

10.Mai.2010

erfolgen und in der Klasse U_Datum in der Form

Mai 10, 2010

Vereinbaren Sie Objekte der Unterklassen, Lesen Sie Datumsangaben unter Verwendung der Lesemethode der Oberklasse ein und geben Sie das Datum mittels der Unterklassenmethoden wieder aus. Weisen Sie dem Pointer aus a) hintereinander die Adressen der Unterklassenobjekte zu und rufen Sie die Ausgabefunktion am Pointer auf.

```
#include <stdio.h>
#include <stdlib.h>

class datum
{ protected:
    int jahr;
    char monat[20];
    int tag;

public:
    void lesedatum()
    { printf("Einlesen eines Datums\n");
      printf("Bitte Tag eingeben:");
      scanf("%d",&tag);
      printf("Bitte Monatsname eingeben:");
      scanf("%s",&monat);
      printf("Bitte Jahreszahl eingeben:");
      scanf("%d",&jahr);
    };
    virtual void schreibedatum()
    { printf("Ausgabe eines Datums\n");
      printf("%s %d\n","Jahr:", jahr);
      printf("%s %s\n","Monat:", monat);
      printf("%s %d\n","Tag:", tag);
    };
};

class D_datum : public datum
{ void schreibedatum()
  { printf("Ausgabe eines Datums im deutschen Format\n");
    printf("%d.",tag);
    printf("%s.",monat);
    printf("%d\n",jahr);
  };
};

class U_datum : public datum
{ void schreibedatum()
  { printf("Ausgabe eines Datums im US Format\n");
    printf("%s ",monat);
    printf("%d,",tag);
    printf("%d\n",jahr);
  };
};
```

```

};

int main()
{
    char c = 'x';
    datum dat;
    datum *datapointer;
    U_datum u_dat;
    D_datum d_dat;
    while ((c != 'e') && (c != 'E'))
    { printf("Bitte Kommando eingeben: l=Lesen allg. Datum, a=Ausgabe allg.
Datum, e= Ende\n");
        printf("                u=Lesen U_Datum, d=Lesen D_Datum\n");
        printf("                x=Ausgabe U_Datum an Pointer, y=Ausgabe
D_Datum an Pointer ");
        fflush(stdin);
        c = getchar();
        switch (c)
        { case 'l':
            case 'L': dat.lesedatum(); break;
            case 'a':
            case 'A': dat.schreibedatum(); break;
            case 'u':
            case 'U': u_dat.lesedatum(); break;
            case 'd':
            case 'D': d_dat.lesedatum(); break;
            case 'x':
            case 'X': datapointer = &u_dat;
                      datapointer->schreibedatum();
                      break;
            case 'y':
            case 'Y': datapointer = &d_dat;
                      datapointer->schreibedatum();
                      break;
            default: break;
        }
    }
    system("PAUSE");
};

```