GRABO Klausur WS04/05

Aufgabe1: Fehlersuche (6 Fehler)

```
Interface F{
   Void f(int i);
};
char d;
public class A implements F{
     char c;
     double g(){
          const int i = 0;
          String s = "GRABO";
          C = s[0];
          Return 3.5;
     }
     public void f(){
          float y;
          c = new char ('c');
          y = s();
     }
};
```

Aufgabe 2:

a) Schreiben Sie das in der Vorlesung behandelte Windows-API-Beispiel zur Fensterprozedur („Demo-Anwendung", Seite 3.3-4) um in eine völlig funktionsgleiche Windows Forms Anwendung!

Füllen Sie dazu das Gerüst der Formular Klasse auf der folgenden Seite aus!

Hinweis: Der Einfachheit halber müssen nur Mausklicks rechts unterhalb des Fenstermittelpunkts korrekt behandelt werden vergl. Teil c)

b) Wie verhält sich die Anwendung aus Teil a), wenn Sie das Fenster durch ziehen vergrößern und erneut klicken?
Kreuzen Sie eines der beiden Fälle an.
Die Ecke welche zuvor in der Fenstermitte lag,
[ ] liegt wieder in der Fenstermitte
[ ] liegt nicht mehr in der Fenstermitte

Tragen Sie in den Rahmen ein, welche Änderungen nötig sind, um das nicht angekreuzte Verhalten zu erreichen.

c) Können Sie sich vorstellen, weshalb das Klickgebiet in Teil a) eingeschränkt wurde?
Welches Problem nämlich beim Zeichnen des Rechtecks auf, wenn nicht rechts unterhalb des Mittelpunkts geklickt wird?

d) welche neuen Arten von Klassenelementen führt .NET ein, die im C++-Sprachstandard nicht vorhandenen sind?
(Nur Namen angeben)

Zu Aufgabe 2:

```
__gc class Myform :  public Form{
private:                        // hier benötigte Elementvariable eintragen:



public:                         // Implementierung der Elementfunktion ergänzen:
    MyForm(){
        this -> Size = System :: Drawing :: Size(800,600);
        this ->  Paint +=                                   // Paint Handler verdrahten
        new PaintEventHandler(this,MyForm::Form_Paint);
        this -> MouseDown +=                                // MousDown Handler verdrahten
            new MouseEventHandler(this,MyForm::Form_MouseDown);



    }
    void Form_Paint(Object* pSender, PaintEventArgs* pArgs){



    }
    void Form_MouseDown(Object* pSender, MouseEventArgs* pArgs){



    }
};
```

Aufgabe 3:

Eine Windows-API-Anwendung soll Texteingaben von der Tastatur entgegennehmen und als Zeichenfolge in der linken oberen Ecke des Hauptfensters darstellen, ähnlich wie ein Textfeld. Buchstaben werden dabei aber durch ein Asterisk-Symbol (Multiplikationszeichen `*`) ersetzt. Nach eingeben von 'F', 'e', 'b', '0' und '5' wird also '***05' angezeigt.

- Ergänzen Sie die folgenden Fensterprozeduren sodass Sie das gewünschte leistet!
- Sie soll mit dem Windows-Hauptprogramm aus der Vorlesung zusammenarbeiten.

Hinweis: Eine Navigationinnerhalb der Zeichenfolge (Backspace- oder Cursortasten) oder eine anderweitige Behandlung von Sondertasten ist nicht nötig. Einen Überlauf der Zeichenfolge brauchen Sie ausnahmsweise nicht zu behandeln.

```
LRESULT CALLBACK WndProc (HWND hWnd, UINT wMessage,
                                    WPARAM wParam, LPARAM(Param))
{


    switch (wMessage) {                    // Messageverteilung


        case WM_PAINT: {


        }       // Ende WM_PAINT


        case WM_DESTROY:                   // Code ausgelassen


    }           // Ende switch
    return DefWindowProc (hWnd, wMessage, wParam, lParam);
}
```

Keller Teil:

Aufgabe 1: (Fehlersuche)

Sie sollen eine Java-Standallone-Anwendung mit GUI unter Verwendung von Swing-Komponenten implementieren.

1.1 Beim starten des Programms erscheint zwar das Anwendungsfenster, es ist jedoch nur die Titelseite sichtbar. (2Pkt.)

- Beschreiben Sie die Ursache für den Fehler
- ➢ Korrigieren Sie das Programm im Anhang 2 mit den notwendigen Änderungen

1.2 Nach Korrektur des Programms wird der Fensterinhalt sichtbar. Im oberen Bereich fehlt jedoch der text „FH Ravensburg-Weingarten“ und es fehlt die Menüleiste. (8Pkt.)

- Beschreiben Sie die Ursache für den Fehler
- ➢ Korrigieren Sie das Programm im Anhang 2 mit den notwendigen Änderungen

1.3 Nach Korrektur des Programms erscheint das Fenster korrekt auf dem Bildschirm. Sie wollen die Anwendung über den schließe Knopf in der Fensterleiste beenden, es passiert aber nichts. (2Pkt.)

- Beschreiben Sie die Ursache für den Fehler
- ➢ Korrigieren Sie das Programm im Anhang 2 mit den notwendigen Änderungen

1.4 Sie testen erneut das korrigierte Programm und stellen fest, dass bei Auswahl der Menüpunkte keine Aktionen passieren. (22Pkt.)

- Beschreiben Sie die Ursache für den Fehler
- ➢ Korrigieren Sie das Programm im Anhang 2 mit den notwendigen Änderungen, sodass bei Auswahl der Menüpunkte die beschreibenden Aktionen auch stattfinden.

Aufgabe 2:

2.1 Im Konstruktor der Objektklasse „Studiengangsbaum.java“ finden Sie als erste Anweisung super(root). (2Pkt.)

- Kann man die Anweisung weglassen oder ersetzen?
- ➢ Begründen Sie ihre Antwort.

2.2 In der Klasse FHMenue wird die Objektinstanz der Klasse FHWgt benötigt. Diese wird im Konstruktor per Parameter übergeben. In der Vorlesung wurde die Methode getToplevelAncestor() vorgestellt mit der man das Hauptfenster erfragen kann. (3Pkt.)

- Ist es möglich im Konstruktor der Objektklasse FHMeunue auf den Parameter zu verzichten und stattdessen das Hauptfenster mit der Methode getToplevelAncestor() zu erfragen?
- ➢ Begründen Sie ihre Antwort.

2.3 Im Konstruktor der Objektklasse Adressdialog wird als erste Anweisung die Methode super() mit zwei Parametern aufgerufen. (4Pkt.)

- Was würde passieren wenn diese Anweisung löschen würde. Wird das Adressfenster noch korrekt angezeigt werden?

2.4 In der Objektklasse Adressdialog werden die einzelnen Zeilen der Adresse über Labels dargestellt. Durch das Gridlayout werden die Zeilen untereinander dargestellt. (2Pkt.)

- Kann man durch Verwendung eines anderen LayoutManagers die gleiche Darstellung erreichen?
- ➢ Begründen Sie ihre Antwort.

AdressDialog
Baumloeschen
FHMenue
FHWgt
Kotaktlistener
LoescheStudiengang
Studiengangsbaum