

Friedrich-Schiller-Universität Jena Fakultät für Mathematik und Informatik Institut für Informatik Lehrstuhl für Programmiersprachen und Compiler	Höhere Programmierung SS 2001	Praktikumsblatt 1 Ausgabe: 02.04.2001 Termin: 05.04.2001
--	--	---

Einführung in das Praktikum Höhere Programmierung

Bevor wir mit dem Programmieren starten wollen wir uns mit dem System und der Programmierumgebung befassen.

Aufgabe 1: Einführung in die ObjectAda-Programmierungsumgebung

Mit dieser Aufgabe soll der Umgang mit der Programmierumgebung in ObjectAda geübt werden. Abbildung 1 zeigt die ObjectAda-Programmierungsumgebung. Das linke Fenster ist der Project Workspace, hier werden die Dateien angezeigt, die zu einem Projekt gehören. Das untere Fenster ist das Output-Window, hier werden die Meldungen beim Compilieren des Programms angezeigt. Der große Teil rechts vom Projekt-Workspace ist der eigentliche Arbeitsbereich hier werden die offenen Dateien, die editiert werden können, angezeigt.

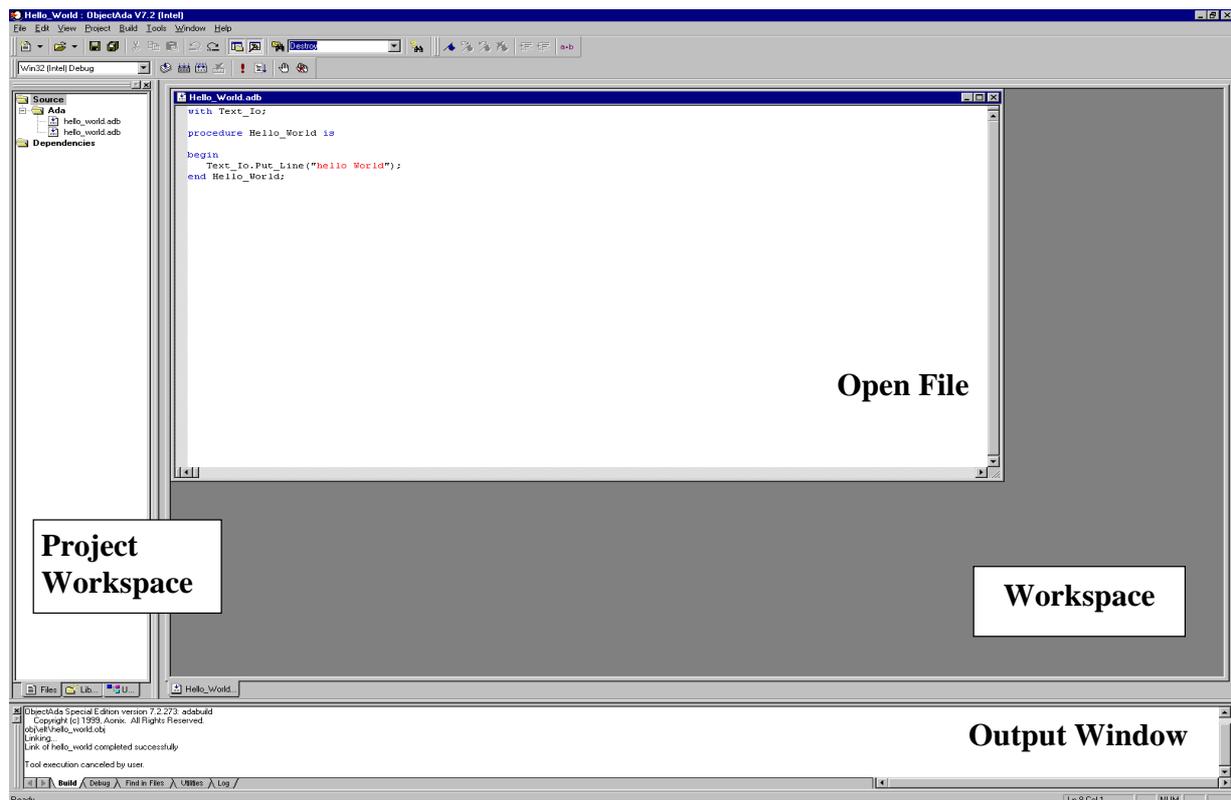


Abbildung 1: Die Oberfläche von ObjectAda mit dem ‚Hello World‘ Programm.

ObjectAda wird vom Startmenü aus gestartet. Das erste Fenster, das erscheint, macht auf die Begrenzung der ObjectAda Demoversion aufmerksam. Diese Limitierung der Demoversion kann für die Dauer unseres Praktikums bedenkenlos mit ‚OK‘ beantwortet werden. So groß werden

unsere Programme sicherlich nicht.

Beim ersten Start sind die Fenster Project-Workspace und Output-Window noch nicht vorhanden. Das Projekt-Workspace Fenster wird erst nach dem Öffnen oder Starten eines Projektes sichtbar.

Bevor wir ein Projekt öffnen sollen hier die wichtigsten Menüpunkte aus der Menüleiste erklärt werden. Die Menüauswahl ‚File‘ enthält alle Funktionen, die mit dem neu Erstellen, Öffnen und Schließen von einzelnen Files und ganzen Projekten zu tun haben. In der Menüauswahl ‚Edit‘ sind alle aus anderen Textverarbeitungsprogrammen wichtigen Funktionen vereint. Als Beispiele seien hier Kopieren, Einfügen, Ausschneiden, Suchen und Ersetzen genannt. Die Menüauswahl ‚View‘ bietet die Möglichkeit, die Fenster Project-Workspace und Output-Window anzuzeigen. Über das Menü ‚Project‘ können wir Dateien zu unserem Projekt hinzufügen. Den Übersetzer und die Funktion des Linkers findet man im Build-Menü. Die anderen Menüpunkte sind selbsterklärend, das Menü Tools werden wir in der zweiten Aufgabe kennenlernen.

Um das erste Programm eingeben zu können, muss ein neues Projekt erstellt werden. Das wird über das Menü ‚File/New Project‘ durchgeführt. Sie werden dann aufgefordert den Projekt Namen einzugeben. Im Feld ‚Plattformen‘ soll das Feld ‚Java‘ keinen Haken haben, dieser muss entfernt werden. Zuletzt muss noch ein Verzeichnis eingegeben werden, wo das Projekt gespeichert wird. Das Feld ‚Insert files upon project creation‘ soll auch keinen Haken haben. Noch sind keine Dateien erstellt, die wir in das Projekt einfügen wollen. Wenn alles ausgefüllt wurde, muss nur noch auf den Button ‚Create‘ gedrückt werden, um das Projekt zu erstellen.

Jetzt erscheint das Project-Workspace-Fenster auf der linken Seite. Mit dem Menüpunkt File/New kann jetzt eine neue Datei eröffnet werden. Diese Datei trägt noch den Namen ‚Unbenannt‘. Eine Datei wird gespeichert mit dem Menüpunkt ‚File/Save‘. Bei einer neu erstellten Datei wird dann nach einem Namen gefragt. Eine Ada-Datei sollte heißen wie der darin befindliche Programmbaustein (Prozedur/Paket).

Als erstes Programm soll das berühmte ‚Hello World‘ Programm in Ada erstellt werden. Zunächst hier das Programm:

```
with Ada.Text_IO;

procedure Hello_World is
begin
    Ada.Text_IO.Put_Line("Hello World");

end Hello_World;
```

Das Programm besteht aus einer Prozedur, der Hauptprozedur. Die Datei, in der das Programm gespeichert wird, sollte denselben Namen haben wie die Hauptprozedur des Programms, also hier „Hello_World“. Das Schlüsselwort `procedure` gibt den Beginn einer Prozedur an. Nach dem Schlüsselwort `is` beginnt der Deklarationsteil und mit `begin` fängt der Anweisungsteil der prozedur an. Mit dem Schlüsselwort `end` wird das Ende der Prozedur angezeigt. Dem Schlüsselwort `end` kann zu Dokumentationszwecken der Prozedurname folgen, was empfohlen wird.

Mit der `with` Anweisung werden andere Pakete eingebunden werden. Die wichtigsten Pakete für die Standard E/A sind `Ada.Text_IO`, `Ada.Integer_Text_IO` und `Ada.Float_Text_IO`.

Nun wollen wir das Programm übersetzen (compilieren). Dazu muss die Datei erst in das Projekt einfügen. Das erfolgt über ‚Project/Add Files‘, dann die Datei auswählen und einfügen. Der Ü-

bersetzungsvorgang wird durch den Menüpunkt ‚Build/Compile‘ gestartet. Werden nach dem Übersetzen keine Fehler angezeigt, kann das eigentliche ausführbare Programm zusammengesetzt (gelinkt) werden. Das wird mit den Menüpunkt ‚Build/Build‘ ausgeführt, und mit ‚Build/Execute‘ wird das fertige Programm dann ausgeführt.

Für die gängigsten Befehle und Funktionen der Programmierumgebung gibt es auch Icons in der Iconleiste. Diese sind meistens selbsterklärend und können durch das Pop-up-Hilfesystem beim Verweilen des Mauszeigers über dem Icon erkannt werden.

Aufgabe 2: Erstellen einer grafischen Benutzungsoberfläche

In der Aufgabe 1 wurde ein einfaches Ada Programm erstellt. Das Ziel dieser Aufgabe ist es, eine Grafische Benutzungsoberfläche zu erstellen. In ObjectAda ist zu diesem Zweck der GUI-Builder als Tool enthalten. Es kann von der Programmierumgebung direkt gestartet werden. Der GUI-Builder wird durch den Menüpunkt Tools/GUI-Builder gestartet.

Beim Start des Tools wird eine Nachricht angezeigt. Wir können also nur 2 Fenster mit jeweils 4 Objekten erstellen. Für unser ‚Hello World‘-Programm mit grafischer Benutzungsoberfläche soll das genügen.

Die Oberfläche startet auch gleich mit einem offenen Fenster, mit dem wir arbeiten können. Es ist mit Window1 beschriftet. Zusätzlich ist der GUI Builder – Editor, in dem der Programmtext erscheint, und der Project-Manager, mit zwei Tabs Properties und Windows, geöffnet. Als drittes kann noch der Application-Builder offen sein, dieser wird für das Übersetzen der grafischen Oberfläche benötigt. Abbildung 2 zeigt den GUI-Builder.

Zuerst soll wieder die Umgebung des GUI-Builder erklärt werden. Das Menü File enthält dieselben Unterpunkte wie auch das ObjectAda ‚File‘-Menü. Durch das Menü ‚Edit‘ kann kopiert, ausschnitten und eingefügt werden. Mit dem Menü ‚View‘ können die Fenster Project-Manager, Editor, Application-Builder und der Menu-Designer angezeigt werden. Mit dem ‚Insert‘-Menü können neue Fenster in das Projekt eingefügt werden. Soll ein Fenster wieder aus dem Projekt gelöscht werden, so wird dies durch den Menüeintrag ‚File/Remove-Current-Window‘ ausgeführt.

Mit dem Menü ‚Options‘ kann ein Gitter angezeigt werden und für ein einfacheres Arbeiten beim Erstellen das Ausrichten an dem Gitter eingeschaltet werden. Auch können verschiedene Projekteinstellungen vorgenommen werden.

Mit dem ‚Help‘-Menü kann die Online Hilfe angezeigt werden. Um sich mit den verschiedenen Parametern und Funktionen der grafischen Benutzungsoberfläche vertraut zu machen wird empfohlen, diese etwas genauer zu studieren.

Um Elemente der grafischen Benutzungsoberfläche hinzuzufügen, befindet sich in der unteren Iconreihe verschiedenen Icons, welche die verschiedenen möglichen Objekte darstellen. Das erste Icon stellt den ‚Pointer-Modus‘ ein. Damit kann die Platzierung und Größe der vorhandenen Objekte verändert werden. Die folgenden Icons stellen die verschiedenen Elemente dar. Die Icons sind größten Teils selbsterklärend.

Die Aufgabe ist, eine grafische Benutzungsoberfläche zu erstellen, die ein Textfeld für die Anzeige von Text enthält, einen Button „Show“ um in diesem Textfeld „Hello World“ anzuzeigen und einen Button „Clear“ um den Inhalt des Textfeldes zu löschen.

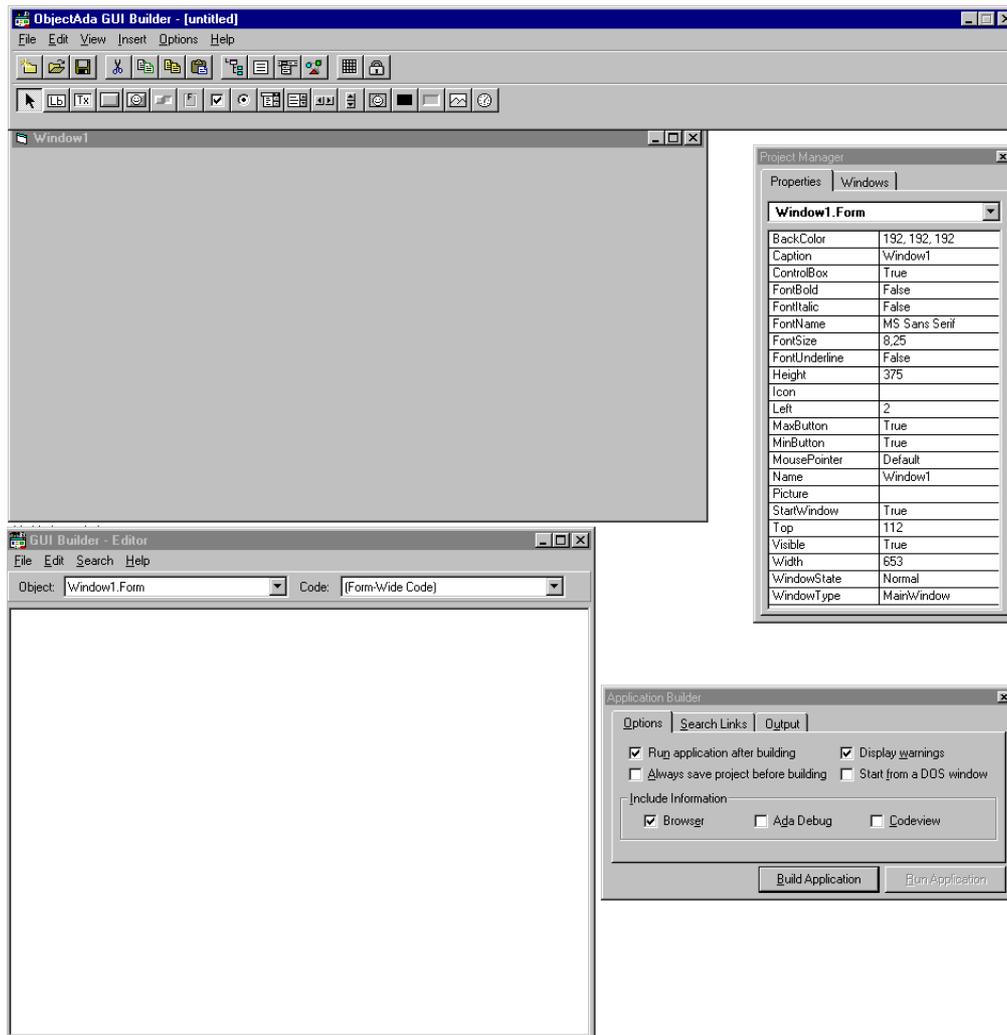


Abbildung 2: Programmierumgebung des GUI Builders.

Zuerst wird das Textfeld erstellt. Dafür wird das Icon ‚Text-Box-Control‘ benötigt. Einfach anklicken und dann mit dem Mouse-Pointer den Bereich, den das Textfeld einnehmen soll markieren. Markiert wird indem die Position einer Ecke im Fenster angeklickt und den linken Mouse-Button gedrückt halten. Jetzt kann der Bereich markiert werden. Im Project-Manager wird sofort ein Objekt Text1 erstellt. Für die Buttons wird das Icon ‚Button-Control‘ benötigt.

Nachdem die gewünschten Objekte eingefügt wurden, werden jetzt einige Eigenschaften der Objekte erklärt. Viele Eigenschaften des Textfeldes sind selbsterklärend, wie ‚Alignment‘, ‚BackColor‘ und ‚ForeColor‘. Die Eigenschaft ‚Name‘ bezeichnet den Objektname, während die Eigenschaft Text den Inhalt bezeichnet. Mit ‚ReadOnly‘ kann gesetzt werden, ob das Objekt vom Benutzer verändert werden kann, in diesem Fall bedeutet ‚False‘, dass der Benutzer Zugriff hat und selbst in das Textfeld schreiben kann.

Die Eigenschaften des Objektes Button sind auch größtenteils selbsterklärend. Die Eigenschaft ‚Caption‘ zeigt die Beschriftung des Buttons. Auch hier ist der Name der Objektname unter dem das Objekt im Programm selbst definiert ist. Mit ‚Enabled‘ kann der Button aktiviert oder deaktiviert werden. Als Beispiel könnte der Button zum Löschen deaktiviert sein, sollte das Textfeld bereits leer sein.

Spielen Sie ein wenig mit den Attributen im Project Manager, damit Sie ein Gefühl für die Erstellung von grafischen Oberflächen bekommen. Die meisten Attribute (Properties) sind selbsterklärend.

Der Programmcode für die Buttons wird im GUI-Builder – Editor Fenster eingegeben. Wichtig sind dabei die beiden Felder ‚Object‘ und ‚Code‘. Im Feld ‚Object‘ wird das Objekt, das eine Aktion ausführen soll, ausgesucht. Im Feld ‚Code‘ wird das Ereignis angegeben, durch das die Aktion ausgelöst werden soll. Im Falle des ‚Hello World‘ Programms ist es das Ereignis ‚Click‘. Es wird im unteren Teil des Fensters bereits der Anweisungsteil der entsprechenden Prozedur angezeigt.

Nun wird der Code eingefügt. Um den Inhalt eines Textfeldes zu setzen oder abzufragen, werden die Funktion und Prozedur `Text` benötigt. Die Prozedur `Text` hat zwei Parameter, der erste Parameter gibt das Objekt an, das eine Text-Eigenschaft hat, der zweite Parameter gibt den neuen Inhalt des Textfeldes an. Die Funktion `text` hat nur einen Parameter, der das Textfeld angibt. Es liefert den momentanen Inhalt des Textfeldes zurück. Für den Button „Show“ brauchen wir den Prozeduraufruf `Text(Window1.<Name des Textfeldes> "Hello World");` und für den Button „Clear“ den Prozeduraufruf `Text(Window1.<Name des Textfeldes> "");`

Nun muss das Programm noch übersetzt und ausgeführt werden. Sollte das Fenster ‚Application-Builder‘ noch nicht offen sein, so kann es über das Untermenü ‚View/Application Builder‘ geöffnet werden. Mit dem Button ‚Build Application‘ im unteren Bereich des Application-Builder kann die Anwendung übersetzt und zusammengesetzt werden (compiled und built). Nach dem dieser Prozess beendet ist, sollte die Anwendung von selbst starten, wenn nicht, dann sollte zumindest der Button ‚Run Application‘ jetzt aktiviert sein. Sollte dies nicht der Fall sein, so könnten beim Übersetzen Fehler aufgetreten sein, diese werden in dem Textbereich des Application Builder, unter dem Abschnitt ‚Output‘ im Application-Builder, angezeigt.

Weiterführende Informationen über den GUI-Builder sind in der Onlinehilfe zu finden.

Viel Spaß beim Ausprobieren.