Erfolgsbedingungen für Anfänger zum Erlernen der Programmierung portabler GUIs in einem Semester

Rony G. Flatscher [1] und Till Winkler [1]

Abstract: An der Wirtschaftsuniversität Wien können Bachelor-Studierende objektorientiertes Programmieren von Grund auf lernen und sind unter anderem in der Lage, portable Programme mit grafischen Benutzeroberflächen für Windows, macOS und Linux zu entwickeln. Dies ist in nur einem Semester, innerhalb von 60 Präsenzstunden mit einem Gesamtlernaufwand von 200 Stunden – 8 European Credit Transfer System (ECTs) – möglich. Dieser Workshop befasst sich mit den Aspekten, die dies möglich machen, einschließlich der Kursziele, des Kursmaterials, der angewandten pädagogischen Prinzipien und der unterrichteten Programmiersprache. Die Teilnehmende des Workshops werden in der Lage sein, den Kurs selbst zu unterrichten.

Keywords: Programmierausbildung, GUIs, ooRexx, Message Paradigma, Java

1 Einleitung

Im Rahmen eines seit mehr als 35 Jahren laufenden Lehrprojekts wurde ein Einführungskurs entwickelt, der es Anfängern ermöglicht, die Grundlagen objektorientierter Programmierung zu erlernen und die erworbenen Fähigkeiten und Kenntnisse in Unternehmen anzuwenden. Die Studierenden werden befähigt, MS-Office-Anwendungen zu automatisieren, portable GUIs und Client/Server-Socket-Anwendungen zu entwickeln sowie XML-Verarbeitung, Web-Scraping und Java-Klassen wie Jsoup (u.a.) zu nutzen [FM21].

Durch teilnehmende Beobachtung, ständige Analyse der Schwierigkeiten (und Erfolge) der Studierenden und Unterrichtsexperimente (z.B. Verwendung verschiedener Programmiersprachen in verschiedenen Semestern, Variation des Inhalts, Identifizierung des kleinsten erforderlichen Wissenselements u.ä.) wurde mit der Zeit das Ziel objektorientierten Programmierung in nur einem Semester zu vermitteln erreicht. Ein wesentlicher Erfolgsfaktor ist dabei das „Message“-Paradigma von ooRexx, denn es erleichtert den Studierenden den Umgang mit komplexen Umgebungen und ermöglicht ihnen die problemlose Nutzung von verschiedenen Paketen (wie BSF.CLS und UNO.CLS) und Bibliotheken (wie BSF4ooRexx850, eine Java-Bridge) [WF24].

Im Einführungskurs „Business Programming“ wenden die Studierenden in der ersten Hälfte des Semesters die erlernten Konzepte auf Windows und MS Office an. Danach folgt die

1 Wirtschaftsuniversität Wien, Department of Information Systems and Operations Management, Welthandelsplatz 2, 1020 Wien, Österreich,
rony.flatscher@wu.ac.at, https://orcid.org/0000-0003-4881-7215;
till.winkler@wu.ac.at, https://orcid.org/0000-0002-4045-3085

doi:10.18420/se2025-ws-10

Nutzung der Java-Laufzeitumgebung, die Erstellung portabler GUIs, Client/Server-Sockets-Programme, XML-Verarbeitung und die Interaktion mit OpenOffice und LibreOffice.

2 Ziele, Materialien und Ablauf des Workshops

Ziel dieses Workshops ist es, alle notwendigen Konzepte und Kenntnisse zu vermitteln, um den Einführungskurs „Business Programming“ unterrichten zu können. Alle Materialien wie Folien, Beispiele sowie Pakete und Bibliotheken sind Open Source und können sofort genutzt werden [Fl24a; Fl24b]. Der Workshop ist für vier Stunden angesetzt und besteht aus vier verschiedenen Teilen

2.1 Lernergebnisse, Programmiersprache und Ablauf

Die Lernergebnisse bestehen hauptsächlich darin, grundlegende Programmierkonzepte zu vermitteln und ihre Anwendung anhand kleiner praktischer Beispiele zu demonstrieren. Eine Einführung in die Theorie der kognitiven Belastung ermöglicht es zu entscheiden, welche Programmiersprachen für Anfänger am besten geeignet sind. Dies muss nicht unbedingt eine beliebte Programmiersprache wie C++, Java oder Python sein, da statische und streng typisierte Sprachen einen höheren intrinsischen kognitiven Aufwand für Anfänger bedeuten können [WF23]. ooRexx ermöglicht es, grundlegende Programmierkonzepte schnell zu vermitteln, da eine geringere kognitive Anforderung notwendig ist. Mit dem Ziel, den extrinsischen kognitiven Aufwand zu reduzieren, wurde ein spezifischer Ablauf des Kurses und Regeln für dessen Durchführung entwickelt. Um die kognitive Belastung zu verringern, wurde die Reihenfolge der Inhalte, die Folien selbst und darin verwendete Nutshell-Beispiele ständig evaluiert und verbessert. Die Studierenden sollten so weit wie möglich von intrinsischer und extrinsischer kognitiver Belastung befreit werden, damit die Kapazität des Arbeitsgedächtnisses für das Lernen genutzt werden kann.

2.2 Programmierkonzepte (ooRexx), Windows OLE und Java-Laufzeitumgebung

Die dynamische und dynamisch typisierte Programmiersprache REXX wurde speziell entwickelt, um leicht erlernbar zu sein, einfach mit Befehlen zu arbeiten und um komplexe Anwedungen umsetzen und pflegen zu können. Im Sinne des Human-Centered-Designs vermitteln Anweisungen ihre Bedeutung in einfacher englischer Sprache, so dass REXX-Programme wie Pseudocode gelesen werden können. ooRexx, der objektorientierte Nachfolger von REXX, folgt diesem Designprinzip und hält daher die Syntax einfach und führt das „Message“-Paradigma ein, welches es ermöglicht, die Verwendung von Objekten vom Implementierungskontext zu abstrahieren, indem Nachrichten für die Interaktion verwendet werden. Nur diese Art der Interaktion, das Senden von Nachrichten, ermöglicht es Anfängern, komplexen Aufgaben schnell zu meistern.

Die Windows-Version von ooRexx wird mit einer Proxy-Klasse (OLEObject) ausgeliefert, welche die Interaktion mit Windows-Objekten (über COM und OLE) auf einen bloßen Austausch von Nachrichten abstrahiert. So können auch Anfänger Programme erstellen, die mit jedem Windows-Objekt über OLE interagieren. Über 20 Jahren hinweg hat der Erstautor eine Java-Brücke entwickelt und verbessert, die die Interaktion mit Java über eine Proxy-Klasse (BSF) ermöglicht [Fl23]. Diese erlaubt die Interaktion mit Java-Objekten durch den Austausch von Nachrichten, wodurch Studierende mit geringem Aufwand Java nutzen können.

2.3 Zentrale Lektionen

Die Folien und darin enthaltenen Beispiele wurden kontinuierlich verbessert und können leicht verstanden und sofort für den Programmierunterricht eingesetzt werden. Es gibt jedoch wichtige Lektionen, die für den Erfolg des Kurses entscheidend sind. Ein Beispiel ist die Einführung in die Definition und Verwendung von Klassen, welche überwältigend sein kann und daher Zeit zum Verarbeiten, Verstehen und Anwenden erfordert.

2.4 Hands-on: Software installieren und Nutshell Beispiele

In diesem Teil des Workshops wird die zu installierende Software und das Ausführen der Nutshell-Beispiele behandelt. Je nach Vorliebe können während des Workshops auch die portablen Versionen von ooRexx ausprobiert werden.

Literaturverzeichnis

[Fl23] Flatscher, R. G.: Proposing ooRexx and BSF4ooRexx for Teaching Programming and Fundamental Programming Concepts. Proceedings of ISECON 2023, 2023.

[Fl24a] Flatscher, R. G.: Business Programming Slides, Part 1 (first half of semester, from zero programming to Windows, MS Office), Vienna University of Economics and Business, 2024, URL: https://wi.wu.ac.at/rgf/wu/lehre/autowin/material/foils/.

[Fl24b] Flatscher, R. G.: Business Programming Slides, Part 2 (second half of semester, BSF4ooRexx camouflaging Java as ooRexx), Vienna University of Economics and Business, 2024, URL: https://wi.wu.ac.at/rgf/wu/lehre/autojava/material/foils/.

[FM21] Flatscher, R. G.; Müller, G.: "Business Programming-Critical Factors from Zero to Portable GUI Programming in Four Hours. In: 6th Business and Entrepreneurial Economics 2021-Conference Proceedings. S. 76–82, 2021.

[WF23] Winkler, T.; Flatscher, R. G.: Cognitive Load in Programming Education: Easing the Burden on Beginners with REXX. In: Central European Conference on Information and Intelligent Systems. Faculty of Organization und Informatics Varazdin, S. 171–178, 2023.

[WF24] Winkler, T.; Flatscher, R. G.: Employing the Message Paradigm to Ease Learning Object-oriented Concepts and Programming. In: 2024 47th MIPRO ICT and Electronics Convention (MIPRO). IEEE, S. 1244–1249, 2024.