

Programmieraufgaben auf PABS bearbeiten

Tim Hegemann

Alexander Dallmann

9. August 2019

PABS steht für Programmieraufgaben Bewertungssystem und wird an der Universität Würzburg mit dem Ziel entwickelt, Studierende beim Erlernen der Grundtechniken des Programmierens und einiger im Programmieralltag notwendiger Tools zu unterstützen. Für einige Veranstaltungen werden auf PABS Programmieraufgaben bereitgestellt, zu denen Sie in einer Entwicklungsumgebung Ihrer Wahl Lösungen entwickeln. Mit PABS können diese dann auf Korrektheit geprüft werden. Findet PABS Fehler in Ihrer Lösung, hilft Ihnen das generierte Feedback dabei, diese zu finden und zu korrigieren.

1 Voraussetzungen

Bevor Sie mit der Bearbeitung der Programmieraufgaben beginnen können, müssen einige Voraussetzungen geschaffen und Programme eingerichtet werden.

1.1 Zugang zu PABS

PABS ist nur im Netzwerk der Universität erreichbar. Wenn Sie an einem der Computerarbeitsplätzen der CIP-Pools der Fakultät für Mathematik und Informatik arbeiten oder in das sichere WLAN der Universität (RZUWsec) eingeloggt sind, ist dies bereits gegeben.

Um auch von zuhause oder unterwegs auf PABS zugreifen zu können, ist es hingegen notwendig, per VPN eine Verbindung zur Universität aufzubauen. Eine Anleitung dafür stellt das Rechenzentrum auf seiner Website bereit¹. Bei Problemen steht Ihnen der IT-Support des Rechenzentrums zur Seite².

1.2 Java Platform, Standard Edition

Für alle derzeit auf PABS verwendeten Programmiersprachen ist die Installation des Java Development Kits (JDK) Voraussetzung. PABS setzt dabei auf die jeweils aktuelle Version des OpenJDK. Auf den meisten Linux Distributionen können Sie dieses direkt aus den jeweiligen Repositories laden. Für alle Betriebssysteme ist auch das weitgehend identische Oracle JDK³ hinreichend.

¹Siehe <https://www.rz.uni-wuerzburg.de/dienste/it-sicherheit/vpn/>

²Siehe <https://www.rz.uni-wuerzburg.de/it-support/>

³Download unter <https://www.oracle.com/technetwork/java/javase/downloads/index.html>

1.3 Integrierte Entwicklungsumgebung

Für die allermeisten Programmiersprachen ist es sinnvoll, eine Integrierte Entwicklungsumgebung (IDE) zu verwenden. Für die Abgabe auf PABS spielt diese per se aber keine Rolle. Sie können also eine beliebige IDE verwenden, die Ihnen gerade angenehm ist und diese auch während der Bearbeitung wechseln.

1.4 Subversion

Der Quellcode der einzelnen Aufgaben wird in einem Apache Subversion (SVN) Repository verwaltet. Für jeden Kurs existiert ein Repository in dem jeder Student einen Unterordner bekommt, auf den nur er zugreifen kann. Der Zugriff auf das Repository erfolgt mit Hilfe eines Clients. Es existiert eine Vielzahl von Clients und die meisten Entwicklungsumgebungen haben entweder einen Subversion Client integriert oder er kann als Plugin installiert werden.

Eine Einführung in die Versionsverwaltung mit Subversion würde den Rahmen der Anleitung sprengen, daher wird hier darauf verzichtet und auf die ausführliche Anleitung des Herstellers⁴ verwiesen.

Nachfolgend wird die Installation eines command-line Clients für gängige Betriebssysteme geschildert, wie er im folgenden in dieser Anleitung verwendet wird.

Linux

Auf allen größeren Linux Distributionen kann SVN über den Paketmanager installiert werden. Einer der Befehle `sudo apt install subversion`, `sudo dnf install subversion` oder `sudo pacman -S subversion` sollte dem Nutzer bekannt vorkommen.

Windows

Für Windows gibt es verschiedene Distributoren, die Installationsarchive für Subversion zur Verfügung stellen. Wir empfehlen die Verwendung von SlikSVN⁵. Achten Sie darauf, die für Ihre Plattform passende Version (meist 64-Bit) herunterzuladen. SlikSVN kommt als ZIP-Archiv in dem sich eine Installationsdatei befindet. Der Installer leitet Sie durch den Installationsvorgang.

MacOS

Die wohl einfachste Möglichkeit ein aktuelles SVN auf MacOS einzurichten ist, es über den Paketmanager Homebrew⁶ zu installieren. Ist Homebrew auf dem System eingerichtet, lässt sich die Installation von SVN mit dem Befehl `brew install svn` anstoßen.

2 PABS

PABS ist ausschließlich aus dem Uni-Netz heraus unter der Adresse <https://pabs.uni-wuerzburg.de> erreichbar.

⁴Online verfügbar unter <http://svnbook.red-bean.com/>

⁵Download unter <https://sliksvn.com/download/>

⁶Siehe <https://brew.sh/>

2.1 Registrierung und Login

Um PABS nutzen zu können, müssen Sie sich bei PABS anmelden. Die Registrierung erfolgt automatisch mit der ersten Anmeldung. Dazu benötigen Sie den Login und das Passwort des JMU-Accounts (S-Nummer), den Sie vom Rechenzentrum erhalten haben. Das Passwort wird von PABS nicht gespeichert, sondern über den zentralen Verzeichnisdienst des Rechenzentrums überprüft.

2.2 In Kurse einschreiben

PABS wird von mehreren Veranstaltungen gleichzeitig verwendet. Daher müssen Sie sich zunächst in den entsprechenden Kurs einschreiben, bevor Sie mit der Bearbeitung der Aufgaben beginnen können. Klicken Sie dazu links oben auf das *Courses*-Menü, wie in Abbildung 1 gezeigt.



Abbildung 1: Das Kursmenü.

All Courses

- 2014
 - Summer
 - Einführendes Programmierpraktikum - [Subscribe](#)
 - Programmierpraktikum Informatik
 - Vorkurs - [Subscribe](#)
- 2013
 - Summer
 - Programmierpraktikum Informatik (Herbst)
 - Vorkurs - [Subscribe](#)
 - Winter
 - Algorithmen & Datenstrukturen (Informatik) - [Subscribe](#)
 - Algorithmen & Datenstrukturen (WiInf)
 - AssignmentsTest - [Subscribe](#)
 - Einführung in die Informatik - [Subscribe](#)
 - Programmierpraktikum Informatik - [Subscribe](#)
 - Programmierpraktikum WiInf/MCS
 - Programmierpraktikum WiInf/MCS (Frühjahr) - [Subscribe](#)

Abbildung 2: Die Übersicht über alle Kurse.

Das Menü besteht aus zwei Abschnitten. Im oberen Abschnitt finden Sie Links zu allen Kursen, in die Sie sich bereits eingeschrieben haben. Die Kurse sind nach Jahr und Semester gruppiert. Im unteren Abschnitt finden Sie einen Link zu einer Übersichtsseite aller Kurse, auf der Sie sich in neue Kurse einschreiben können.

Die Kurse auf der Übersichtsseite sind wie im Menü in einer Baumstruktur nach Jahr und Semester sortiert dargestellt, siehe Abbildung 2. Für Kurse in die Sie bereits eingeschrieben sind, finden Sie einen Link zum Kurs, andernfalls einen Link zur Einschreibung. Klicken Sie dazu auf *Subscribe*. Bei einigen Kursen ist ein Passwort für die Einschreibung notwendig, das gegebenenfalls beim Betreuer des Kurses erfragt werden kann.

2.3 Die Kursübersicht

Auf der Startseite des Kurses sieht man eine Liste mit Informationen zu den einzelnen Aufgaben, wie in Abbildung 3 dargestellt. Die Aufgaben sind nach der verbleibenden Bearbeitungszeit sortiert, die auch in der rechten oberen Ecke angezeigt wird. Zusätzlich finden Sie Angaben zu Beginn und Ende der Bearbeitungszeit und der Anzahl der Revisionen, die Sie bereits hochgeladen haben. Außerdem wird der Bearbeitungsstatus der Aufgabe angezeigt und die Aufgabe entsprechend farblich hinterlegt:

Failing

Die Aufgabe ist noch nicht erfolgreich bearbeitet worden. Einige Tests werden noch nicht bestanden.

Success

Die Aufgabe wurde erfolgreich bearbeitet, es wurde aber noch keine Lösung eingereicht.

Solution

Die Aufgabe wurde erfolgreich bearbeitet und eine der korrekten Revisionen wurde als Lösung markiert.

Weiterhin finden Sie Links zur Aufgabenstellung (*Instructions*) sowie zur Liste der von Ihnen erzeugten Versionen (*Revisions*) und einen Link zum Subversion Repository für diese Aufgabe.





People's Democratic Dictatorship		No time left	
User:	 ()	Repository Link	Reset Repository
	Instructions / Revisions		
Begin:	22.12.2017 12:00:00	Revisions:	
End:	22.01.2018 12:00:00	Status:	

Abbildung 3: Übersicht zu einer Aufgabe.

2.4 Übersicht über abgegebene Versionen

Wenn Sie dem Link zu den abgegebenen Versionen (*Revisions*) folgen, zeigt PABS Ihnen eine Tabelle mit allen von Ihnen durch das Kommando `svn commit` erzeugten Revisionen. Folgende Informationen zu den Revisionen werden angezeigt:

Revision Die eindeutige Nummer der Revision.

Author Der Benutzer, der diese Revision erzeugt hat.

Timestamp Der genaue Zeitpunkt, zu dem die Revision erzeugt wurde.

State Der Status, in dem sich die Revision befindet.

Message Die beim Commit angegebene Nachricht.

Actions Auf der Revision ausführbare Aktionen (Testen, Details anzeigen oder als Lösung einreichen).

Eine von Ihnen durch einen Commit erzeugte Version durchläuft mehrere Stadien:

NotEvaluated

Mit dieser Revision ist noch nichts geschehen.

EvaluationPending

Diese Revision wird gerade vom System getestet.

Accepted

Die Revision hat alle Tests bestanden.

Rejected

Beim Testen sind in einigen Tests Fehler aufgetreten.

Error

Die Revision konnte nicht getestet werden.

Solution

Die Revision ist aktuell als Lösung für die Aufgabe markiert.

PABS bietet Ihnen für getestete Versionen an, diese als Lösung einzureichen. Dies geht unter *Actions* mit dem Eintrag *Solution*. Je nach Aufgabe ist dies nur für Abgaben möglich, die einen gewissen State erreichen. Es ist weiterhin möglich, eine detaillierte Beschreibung der Testergebnisse abzurufen. Wählen Sie dazu unter *Actions* den Eintrag *Results* aus.

Achtung!

Achten Sie darauf, vor dem Ende des Bearbeitungszeitraums mindestens eine Revision als Lösung zu markieren!

2.5 Testergebnisse

Hat man in der Revisionsübersicht die detaillierten Testergebnisse einer abgegebenen Version aufgerufen, zeigt das System eine Übersicht – ähnlich der in Abbildung 4 gezeigten – mit den Details des Build-Prozesses und den Ergebnissen der einzelnen Tests.

Im oberen Teil sind nochmal die wichtigsten Eigenschaften der Aufgabe und der gewählten Revision aufgeführt. Darunter findet sich der Report, den das System über die Ausführung und Tests der Abgabe erstellt hat. Wie an der Farbe und dem Status *Rejected* erkennbar, erfüllt diese Abgabe nicht alle Anforderungen, da ein Test fehlschlug.

Das Wurzelement des Reports fasst die Meldungen des verwendeten Build Tools – PABS verwendet für Java Aufgaben Gradle – und die Ergebnisse der Tests zusammen. Das Element *Gradle Output* würde wenn aufgeklappt einige Statusmeldungen des Build Tools zeigen, die für einen erfolgreichen Build meist uninteressant sind. Schlägt der Build allerdings fehl, sind hier wertvolle Hinweise für die Fehlersuche zu finden. Das Element *Test Results* enthält die eigentlich interessanten Elemente zu den einzelnen Unit-Tests. Wie die Farbcodierung zeigt, sind die Tests aus der unteren Gruppe *test.java.TestVisitenkarte* bestanden, während der Test *testEinlesen* aus der Gruppe *test.java.TestEinlesenAusgaben* fehlgeschlagen ist. Die angezeigte Fehlermeldung gibt einen Hinweis auf das bemängelte Fehlverhalten. Wie hilfreich diese Meldungen sind, hängt am Einsatz des Aufgabenerstellers. Das Element *Console Output* fasst die Ausgaben des Programms auf die Standard-Ausgabeströme zusammen.

Submission details

Course: Vorkurs Informatik (Summer, 2019) Assignment: Tag3_Visitenkarte User: ██████████ (██████████) Begin: 10.04.2019 09:00:00 End: 11.04.2019 09:00:00	Revision: 59 Message: Wrong order of inputs Timestamp: 10.04.2019 10:44:59 State: Rejected
--	---

Gradle Build Results 0.091s ^

Gradle Output v

Test Results 0.091s ^

test.java.TestEinlesenAusgeben 0.048s ^

testEinlesen 0.048s ^

```
junit.framework.AssertionFailedError: Es konnten nicht alle erwarteten Eingaben des Programms erfüllt werden
Falsche Anzahl, oder die Eingaben wurden über ein statisches Attribut ("Klassenvariable") erwartet
```

Console Output v

test.java.TestVisitenkarte 0.043s v

Abbildung 4: Testergebnisse einer abgegebenen Version.

Jeweils am rechten Rand des Element-Headers findet sich sofern anwendbar eine Anzeige, wie lange der jeweilige Schritt im Build-Prozess gedauert hat. Elemente mit mehreren Unterlemenen aggregieren dabei deren Zeiten. Dauert die Ausführung insgesamt zu lange, wird sie vom System abgebrochen und eine entsprechende Fehlermeldung ausgegeben.

3 Programmieraufgaben bearbeiten

Um eine Programmieraufgabe zu bearbeiten verbinden Sie sich mit dem SVN von PABS, erstellen die geforderten Sourcecodes und synchronisieren diese anschließend mit PABS, sodass die im vorigen Kapitel beschriebenen Aktionen angestoßen werden können.

3.1 Erstellen der Arbeitskopie

Navigieren Sie in den Pfad, wo die Arbeitskopie Ihres Repositories angelegt werden soll. Das folgende Kommando erzeugt die lokale Arbeitskopie.

```
cd <path to your working copy>
svn checkout <repository link>
```

Den Repository Link finden Sie in PABS in der Aufgabenübersicht unter *Repository Link*. In den meisten Browsern können Sie ihn mit *Rechtsklick* → *Copy Link Location* in die Zwischenablage kopieren.

Die lokale Arbeitskopie enthält das Projektverzeichnis, das PABS für Sie für die gewählte Aufgabe angelegt hat und in dem alle Dateien liegen müssen, die von PABS ihren Abgaben für diese Aufgabe zugeordnet werden sollen.

3.2 Sourcecode Dateien anlegen

Unter Umständen hat der Aufgabenersteller für diese Aufgabe ein bestimmtes Format vorgesehen oder bereits ein Template angelegt, das Sie einfach verwenden können. Konsultieren Sie im Zweifelsfall die Aufgabenstellung.

Ansonsten erstellen Sie im dem im vorigen Schritt erstellten Projektverzeichnis den Ordner *src* für Ihre Sourcecodes und bei Bedarf den Ordner *ressources* für andere Dateien, sofern Ihr Programm sie zur korrekten Funktion benötigt.

In der Regel können Sie das Projektverzeichnis nun in die IDE ihrer Wahl importieren und mit dem Programmieren beginnen.

3.3 Eine Version auf PABS abgeben

Wenn Sie mit der Implementierung der Aufgabe fertig sind und Ihre lokalen Tests die korrekte Funktionsweise des Programms bestätigen, können Sie diese Version auf PABS hochladen und dort bewerten lassen.

Fügen Sie dafür der Versionsverwaltung mit folgenden Kommandos alle neu erstellten Sourcecode-Dateien hinzu und prüfen Sie, ob alle für das Programm wichtigen Dateien berücksichtigt wurden.

```
svn add <source file> <another source file> <...>
svn status
```

Der Befehl `svn status` listet dann alle neuen und geänderten Dateien wie im folgenden Beispiel auf:

```
A    src/jpp/addressbook/gui/AddressBookApplication.java
M    src/jpp/addressbook/gui/ContactInfoView.java
A    src/jpp/addressbook/gui/ContactViewModel.java
M    src/jpp/addressbook/gui/FilterPane.java
D    src/jpp/addressbook/gui/Main.java
```

Die Buchstaben der ersten Spalte zeigen dabei die Art der Änderung an. **A** für neu hinzugefügte, **M** für geänderte und **D** für gelöschte Dateien. Ein Ausrufezeichen **!** kennzeichnet eine Datei, die der Versionsverwaltung noch nicht hinzugefügt wurde.

Der `commit`-Befehl synchronisiert diese Änderungen dann mit dem PABS-Server:

```
svn commit -m 'Some changes to the GUI'
> Adding      src/jpp/addressbook/gui/AddressBookApplication.java
> Sending     src/jpp/addressbook/gui/ContactInfoView.java
> Adding      src/jpp/addressbook/gui/ContactViewModel.java
> Sending     src/jpp/addressbook/gui/FilterPane.java
```

```
> Deleting      src/jpp/addressbook/gui/Main.java
> Transmitting file data .....done
> Committing transaction...
> Committed revision 66892.
```

Die so erstellte Revision taucht danach in der Übersicht über abgegebene Versionen, siehe Abschnitt 2.4, in PABS auf. Achten Sie darauf, mit dem Argument `-m` eine aussagekräftige Commit-Meldung anzugeben, damit Sie Ihre Änderungen später nachvollziehen können.

4 Häufig gestellte Fragen

Die PABS-Website ist nicht erreichbar

Überprüfen Sie, dass Sie auch wirklich im Uni-Netz sind. Ihre externe IP sollte den Präfix **132.187** haben. Überprüfen Sie gegebenenfalls, dass das VPN ordnungsgemäß funktioniert und aktiviert ist. Informieren Sie sich bei der entsprechenden Veranstaltung, zum Beispiel in einem passenden Forum, ob eine technische Störung des Systems vorliegt.

Ich erhalte die Meldung „The build timed out“

Jeder Build-Vorgang auf PABS ist zeitlich beschränkt. Diese Beschränkung betrifft auch die Ausführung der Tests. Wird eine bestimmte Ausführungszeit überschritten, bricht das System den Vorgang ab und lässt die nächste Abgabe zum Zuge kommen. Suchen Sie in Ihrem Code nach Endlosschleifen oder Stellen, die durch suboptimale Implementierung sehr ineffizient arbeiten.

Ich möchte auf eine frühere Version der Abgabe zurückkehren

Das ist in SVN nicht so einfach. Sie müssen einen Commit erstellen, der alle Änderungen seit dem gewünschten Zielcommit rückgängig macht und diesen an die bisherige Historie anhängen. Um Inkonsistenzen zu vermeiden, ist es wichtig, dass Sie auf dem aktuellen Stand des Servers arbeiten. Suchen Sie zunächst die Revisionsnummer der Version zu der Sie zurückkehren möchten heraus. Folgende Kommandos übernehmen den Rest:

```
svn update
svn merge -r HEAD:<revision> .
svn commit -m 'rolled back to revision <revision>'
```

Ich erhalte vom SVN die Meldung „File out of date“

Sie arbeiten vermutlich auf einer veralteten Revision, sprich die Version Ihrer Arbeitskopie ist älter als die auf dem PABS-Server. Wenn Sie die lokalen Änderungen übernehmen und die auf dem Server verwerfen wollen, können Sie dies mit folgenden Kommandos tun:

```
svn update --accept mine-full
svn commit -m 'Override with my local changes'
```