

# Aufgabenblatt 0

*git und IDEA*

## Wichtige Ankündigungen

- **Erinnerung Prüfungsanmeldung:** (für die meisten: in MOSES / MTS) Deadline ist am **26.05.2023**. Ohne Prüfungsanmeldung können Sie nicht an der Klausur teilnehmen und bekommen keine Prüfungsleistungen angerechnet.
- Registrieren Sie sich unter <https://testing.ml.tu-berlin.de/index.php>. Siehe Aufgabe 1.
- Das Vorlesungsmaterial, die Übungsblätter und die Vorlagen für die Hausaufgaben finden Sie unter <https://git.tu-berlin.de/algodat-rose23/Material.git>, nachdem Sie sich bei uns im git angemeldet haben.
- Alle Übungen sind in Einzelarbeit zu erledigen. Kopieren Sie niemals Code und geben Sie Code in keiner Form weiter. Die Hausaufgaben sind Teil Ihrer Prüfungsleistung. Finden wir ein Plagiat (wir verwenden Plagiatserkennungssoftware), führt das zum Nichtbestehen des Kurses.
- Dies ist eine demo Hausaufgabe. Sie erhalten dafür keine Punkte, aber sie legen die Grundlage, um alle weiteren Hausaufgaben bearbeiten und abgeben zu können und können den Prozess bereits einmal vollständig durchlaufen.

## Abgabe (bis 01.05.2023 23:59 Uhr)

Die folgenden Dateien müssen für eine erfolgreiche Abgabe im git Ordner eingecheckt sein:

### Geforderte Dateien:

Blatt00/Abgaben.txt	Aufgabe 3.5
Blatt00/src/Aufgabe0.java	Aufgabe 3.5

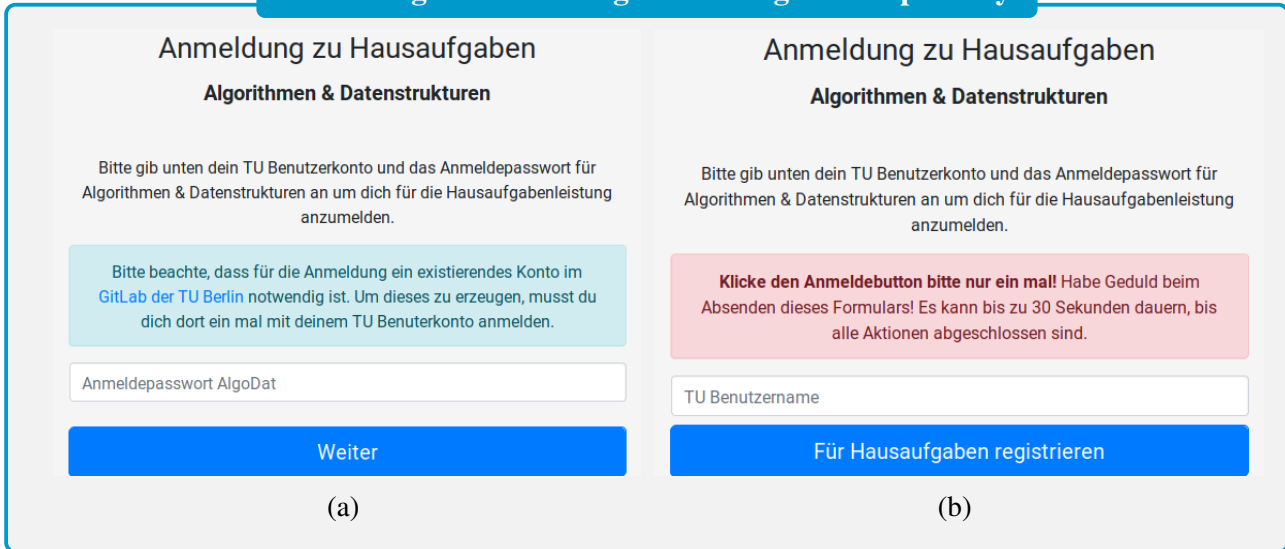
Als Abgabe wird jeweils nur die letzte Version im git gewertet.

## Aufgabe 1: Anmeldung git

Dies ist eine kleine Anleitung zur Anmeldung im **algodat-rose23 git**. Mit der Anmeldung wird für Sie ein eigener Ordner **tuBit-Kennung.git** erstellt. Darüber werden Sie Ihre Hausaufgaben auf unseren Server hochladen. Dort laufen dann die automatischen Tests und Sie erhalten das vorläufige Ergebnis im git.

- 1.1 Wenn Sie sich noch nie in git eingeloggt haben, loggen Sie sich einmal unter <https://git.tu-berlin.de> im tubIT-GitLab ein. Dazu nutzen Sie Ihren tubIT Account. Sollten Sie keinen tubIT Account haben: Schreiben Sie uns schnellstmöglich eine E-Mail an **algodat@neuro.tu-berlin.de**.

Abbildung 1: Anmeldung für das AlgoDat Repository



**Anmeldung zu Hausaufgaben**  
**Algorithmen & Datenstrukturen**

Bitte gib unten dein TU Benutzerkonto und das Anmeldepasswort für Algorithmen & Datenstrukturen an um dich für die Hausaufgabenleistung anzumelden.

Bitte beachte, dass für die Anmeldung ein existierendes Konto im GitLab der TU Berlin notwendig ist. Um dieses zu erzeugen, musst du dich dort ein mal mit deinem TU Benutzerkonto anmelden.

Anmeldepasswort AlgoDat

Weiter

(a)

**Anmeldung zu Hausaufgaben**  
**Algorithmen & Datenstrukturen**

Bitte gib unten dein TU Benutzerkonto und das Anmeldepasswort für Algorithmen & Datenstrukturen an um dich für die Hausaufgabenleistung anzumelden.

Klicke den Anmeldebutton bitte nur ein mal! Habe Geduld beim Absenden dieses Formulars! Es kann bis zu 30 Sekunden dauern, bis alle Aktionen abgeschlossen sind.

TU Benutzername

Für Hausaufgaben registrieren

(b)

- 1.2 Rufen Sie <https://testing.ml.tu-berlin.de/index.php> auf und geben Sie zunächst das AlgoDat-Passwort **AlgoDat2023** ein, um zu Ihrer Anmeldung zu kommen. Siehe Abb. 1 (a)
- 1.3 Geben Sie dann Ihren TU Benutzernamen ein. Siehe Abb. 1 (b).
- 1.4 Installieren Sie git auf Ihrem Computer, wenn Sie das noch nicht getan haben. Eine Anleitung finden Sie u.a. unter den Videos für diese Woche oder z.B. <https://www.git-scm.com/book/de/v2/Erste-Schritte-Git-installieren>.

## Aufgabe 2: IDEA

Wir empfehlen IntelliJ IDEA von *JetBrains* als integrierte Entwicklungsumgebung (IDE) für Java (*free and open source*). Diese wird für Live Demonstrationen in den Videos verwendet. Sie können auch eine andere IDE verwenden, erhalten dann aber keine Unterstützung von den Tutorinnen und Tutoren bei spezifischen Fragen zur Ihrer IDE.

Sie können IDEA Community Version hier herunterladen:

<https://www.jetbrains.com/idea/download/>

- 2.1 Installieren Sie IDEA. Informationen dazu finden sie hier: <https://www.jetbrains.com/help/idea/installation-guide.html> oder unter den Videos für diese Woche. Beachten Sie, dass Sie ebenfalls eine standalone JDK Version installieren müssen. Auf dem Testserver ist die JDK Version 20 installiert. (<https://www.oracle.com/java/technologies/downloads/>)
- 2.2 Erstellen Sie ein erstes Java-Projekt und führen Sie es aus. Für Inspiration oder allgemein zum Üben der Programmierung in Java können sie z.B. den Crashkurs im Material Ordner nutzen.
- 2.3 Programmbeispiele aus der Vorlesung stehen im git (siehe Aufgabe 1) in dem Ordner Material/Code zur Verfügung.

## Aufgabe 3: git

Dies ist eine sehr kurze Einführung in die Nutzung von git. Sie werden im Internet sehr viel aus-

fürlichere Dokumentationen finden. Sie werden dort außerdem zu fast jedem Problem, was sie mit git haben können, eine Lösung finden. Eine Anleitung zur Installation von git finden Sie hier: <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>.

### 3.1 Ein neues Repository herunterladen

Man navigiert in das Verzeichnis, in das man die Arbeitskopie des Repository klonen möchte. Dann:

```
$ git clone /path/to/repository/or/e.g./url
```

Für das Material-Repository ist der Befehl folglich:

```
$ git clone https://git.tu-berlin.de/algodat-bose23/Material.git
```

In Ihrem Verzeichnis **tubIT-Kennung** haben Sie alle Rechte, die Sie brauchen, um Ihre Hausaufgaben hochzuladen. Das Material-Repository können Sie ausschließlich aktualisieren, also die neuste Version herunterladen.

Den Link, den Sie brauchen, um das Repository herunterzuladen, finden Sie auf <https://git.tu-berlin.de>, wenn sie auf das Projekt gehen.

### 3.2 Lokal Änderungen speichern

Um sich ausgeben zu lassen, welche Änderungen im Repository noch nicht hinzugefügt sind:

```
$ git status
```

Änderungen müssen zuerst vorgeschlagen werden mit:

```
$ git add <filename>
```

oder

```
$ git add .
```

um alle Änderungen aufzunehmen. Diese sind dann noch nicht bestätigt. Sie müssen also noch commitet werden:

```
$ git commit -m "Nachricht: Welche Änderung wird commitet"
```

### 3.3 Änderungen hochladen

Nachdem man die Änderungen geaddet und commitet hat, kann man sie so hochladen:

```
$ git push
```

Beachten Sie, dass Ihre Dateien und Änderungen erst nach diesem Schritt mit `git push` und nicht schon nach `commit` auf unseren Server hochgeladen werden. Eine Abgabe ist also erst nach `git push` gültig.

### 3.4 Repository Update

Um ein lokales Repository zu aktualisieren und z.B. die neuen Aufgaben herunterzuladen:

```
$ git pull
```

### 3.5 Aufgabe

Laden Sie sich Ihr eigenes Repository **tubIT-Kennung**, sowie das Material-Repository herunter. Kopieren Sie den Ordner **Blatt00** aus dem Material-Repository nach **tubIT-Kennung**.

Legen Sie dort im Ordner Blatt00 eine txt-Datei mit dem Namen `Abgaben.txt` an. Lesen Sie sich die folgenden Regeln für Ihre Abgaben durch und kopieren Sie den Text in die Datei.

#### Regeln für AlgoDat Abgaben

Ein Paar Regeln für Java Lösungen:

Es darf nicht plagiiert werden. Das heißt, Sie dürfen keine Lösungen anderer für die Bearbeitung der Aufgaben nutzen. Das gilt sowohl für Lösungen aus diesem Semester als auch für Lösungen aus den vorherigen Semestern oder Lösungen, die Sie im Internet gefunden haben. Ein nachgewiesenes Plagiat führt zum Nichtbestehen des Moduls und der Meldung eines Täuschungsversuches an das Prüfungsamt.

Es dürfen Variablen und Methoden zu vorgegebenen Klassen hinzugefügt werden.

Vorhandene Variablen dürfen nicht geändert werden, auch nicht im Typ oder Zugriffsrecht.

Vorhandene Methoden ohne TODO Kommentar dürfen in keiner Weise geändert werden.

Bei vorhandenen Methoden mit TODO Kommentar darf Signatur und Zugriffsrecht nicht geändert werden. Es dürfen allerdings zusätzliche Variablen eingeführt werden.

Es dürfen im Code keine Packages verwendet werden. Diese kompilieren auf dem Server nicht. Das gleiche gilt für imports, die nicht aus der Standard Java Library kommen.

Die vorgegebene Ordnerstruktur in der Abgabe darf nicht geändert werden.

Sie können für schlechte Formatierung des Codes Punktabzug bekommen (da dies die Lesbarkeit und damit die Korrektur erschwert). Um das zu umgehen, können Sie z.B. am Ende Ihren Code mit `Ctrl+Alt+L` (bzw. Code | Reformat Code) formatieren.

Nichtbeachten der Regeln kann zu Punktabzug führen.

Laden Sie den Ordner **Blatt00** einschließlich der Datei anschließend hoch (mit `add`, `commit` und

push wie oben beschrieben). Sie können sich anschließend z.B. auf der gitLab-Website einloggen, sich Ihr Repository ansehen und gucken, ob es geklappt hat.

Das mitgegebene Programm im `src`-Ordner können Sie zu Übungszwecken um eine `main`-Methode erweitern und sich die Datei auf der Konsole ausgeben lassen. Dies ist aber nicht Teil der Aufgabe.

Beachten Sie, dass sie nur in dem **tubIT-Kennung** Repository Änderungen hochladen können, in dem Material-Repository haben Sie 'read-only' Zugriffrechte.

### **Was Sie nach diesem Blatt wissen sollten:**

- wo Sie das Material zum Kurs finden
- wo Sie Ihr eigenes git Repository des Kurses finden
- wie sie ein git Repository herunterladen
- wie sie ein git Repository aktualisieren
- wie Änderungen in Ihrem eigenen Repository speichern und hochladen
- wie sie ein Java-Projekt in IDEA anlegen und ausführen