



Hinweis: Arbeite mit BlueJ 4.x - in der Schule steht eine portable-Version im Ordner public zur Verfügung. Bitte von dort starten

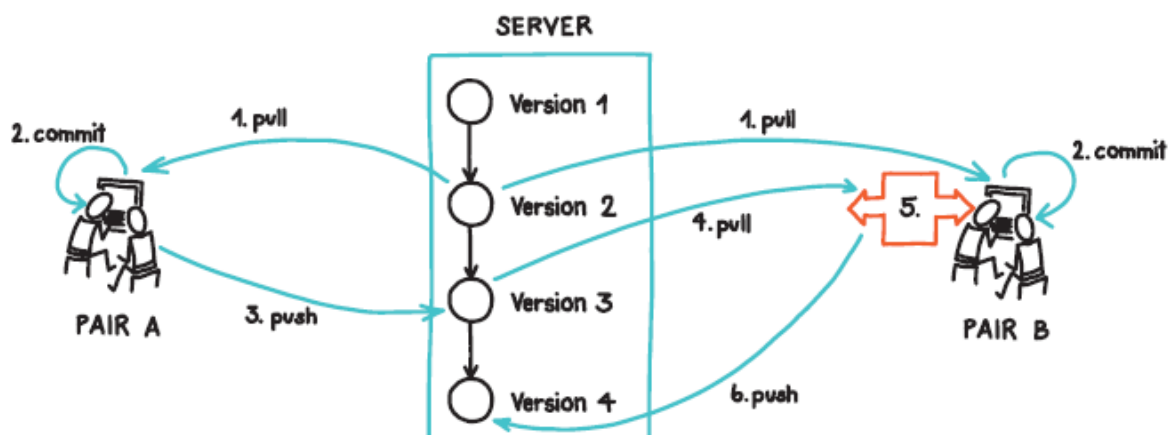
Kollaboratives Arbeiten durch Versionskontrolle unterstützen

Softwareentwicklung ist in der Regel so komplex, dass sie nur im Team entwickelt werden kann. Versionskontrollsysteme unterstützen die verteilte Arbeit in folgenden Punkten

- Verteilter Zugriff auf Dateien
- Versionierung: Übersicht über verschiedene Versionen inklusiv der Dokumentation was, wann geändert wurde
- Datensicherheit, u.a. durch das Rücksetzen der Datei auf eine ältere Version
- Automatisches Zusammenführen (engl. merge) von Quelltexten bzw.

Grundlegende Operationen eines Versionskontrollsystems

- **Checkout – Arbeitskopie erstellen:**
Lokal (auf den eigenen Rechner) alle Inhalte des Versionskontrollsystems (herunter-)laden.
- **Update / Pull – Lokale Daten Aktualisieren:**
Lokal (auf den eigenen Rechner) Änderungen des Versionskontrollsystems (herunter-)laden.
- **Commit – Lokal Änderungen Abgeben:** Eine Änderung auf dem **eigenen** Rechner im Versionskontrollsystem speichern: (ohne Übersetzungsfehler!) Immer in einem kurzen Kommentar die Änderungen beschreiben!
- **Push – Auf dem Server Änderungen Abgeben:** Lokale Änderungen auf den Server übertragen.



Versionskontrolle mit BlueJ: Git Repository unter GitHub

0. Account erstellen

Um auch in ein Repository schreiben zu können muss unter <https://github.com/> ein Account erstellt werden.

[Zur Zusammenarbeit bei einem Projekt erhält man eine Einladung, die man bestätigen muss.]

1. Teamarbeits-Menü einschalten

Bei „Ansicht > Show Team and Test Controls“

ein Häkchen setzen

2. Erstmals ein BlueJ-Projekt aus einem Repository auschecken

„Werkzeuge > Teamarbeit > Arbeitskopie erstellen...“

- Eingabe der Serverdaten (mit Häkchen um die Zugangsdaten für die Zukunft zu speichern)

- (Teste Verbindung)

- Mit OK bestätigen

Zum Speichern

- Speicherort wählen
- Projektname eingeben

3. Nun selbst Programmieren

Im ersten Beispiel bitte selbst eine Spielfigur überlegen und diese als Unterklasse der Klasse SPIELFIGUR umsetzen. (Man kann sich bei den Klassen WOMBAT und BATMAN orientieren.)

- Mindestens zwei Attribute, die sinnvoll im Konstruktor gesetzt werden.
- Zwei Methoden (Eine Methode wird erzwungen, die andere anschauen)

4. Änderungen abgeben über Commit und Push

(siehe Teamwork-Menü in BlueJ)

- a. **Achtung**, falls jemand anderes bereits in das Projekt neuen Quelltext hochgeladen hat, muss man **zuvor den eigenen Stand ein Aktualisieren (Pull)**

- b. Über **Commit** im lokalen Repository speichern.

Wichtig eine knappe Beschreibung, was geändert wurde, z.B. „klasse xy hinzugefügt“ oder „in Methode z wurde ... ergänzt“ (siehe Abbildung rechts)

- c. Über **Push** ins Repository des Servers hochladen.

5. Commits des Teams ansehen

Über den Browser kann man sich Informationen von (öffentlichen) Repositories durch Eingabe der Repository-URL ansehen. Die Änderungshistorie wird unter „Commits“ angezeigt.

