

- (1) Und zwar muss ich dich noch einmal fragen, in welcher Gruppe du warst?
- (2) In der Experimentgruppe.
- (3) Okay, gut. In der Experimentgruppe
- (4) kannst du noch mal kurz erklären, was wir da gemacht haben.
- (5) Wir hatten dieses Experiment, wo uns verschiedene Teile einer, also verschiedene korrekte Teile einer künstlichen Sprache gezeigt wurden und wir danach intuitiv entscheiden sollten, ob andere Sätze in derselben künstlichen Sprache richtig oder falsch sind. Und dann gab es irgendwie noch ein Teil, wo wir schnell reagieren sollten, nachdem welche geometrische Form, in welcher Farbe wir auch schon gesehen haben.
- (6) Ja, okay. Zu deinem sprachlichen Teil, würdest du sagen, dass dir das eher leichter oder schwerer gefallen ist?
- (7) Eigentlich relativ leicht, aber es war nicht alles richtig, also vielleicht so im mittleren Bereich.
- (8) Okay, hast du das Gefühl, dass dir diese Übung irgendwie geholfen hat oder hattest du eher das Gefühl, dass es hinderlich für dich war?
- (9) Also hinderlich auf keinen Fall. Inwiefern es hilfreich war, kann ich glaube ich nicht so gut einschätzen, aber es war auf jeden Fall interessant.
- (10) Ja, okay. Das ist ja jetzt gerade schon ganz gut zusammengefasst, was wir gemacht haben. Kannst du dich noch an einzelne Aspekte dieser Sprache erinnern?
- (11) Es waren immer so ein paar Buchstabenkombinationen, die relativ willkürlich gewirkt haben. Manche davon waren auch ziemlich ähnlich und erst nach einer Weile aufgefallen ist, dass es gar nicht die selben sind, weil die ja nur sehr kurz gezeigt wurden. Also ich glaube, es gab die Kombination a-k (aak) und a-f (aaf) und die sah ja relativ ähnlich aus.
- (12) Ja, das stimmt. Kannst du dich an noch mehr Wörter erinnern bei der Kombination?
- (13) Ich glaube, es gab noch p-r-e-z und g-l-o-x (plox). g-u-m war glaube ich mit dabei und bei dem Rest bin ich mir nicht mehr so sicher. Es gab auf jeden Fall immer so bestimmte Wortkombinationen, die als Wörter bezeichnet haben, die dann hintereinander standen relativ häufig. Also das irgendwie den Eindruck gemacht hat, dass sie irgendwie zusammengehören oder in einer bestimmten Reihenfolge irgendwie vorkommen konnten.
- (14) Und an welche davon kannst du dich besonders gut erinnern?
- (15) Also das, was ich vorhin gesagt habe mit a-a-k oder a-f, die standen immer entweder am Anfang oder nach diesem Teil in der Mitte mit so einem Wort, das glaube ich vier Buchstaben beginnt mit p, also es war dieses p-r-e-z glaube ich und es gab noch ein anderes, was dem relativ ähnlich war. Und dann gab es zwei Wörter, die dann direkt dann nachkommen konnten und nochmal zwei, die dann zum Teil als drittes kamen, wobei das in der Mitte manchmal auch weggelassen wurde.
- (16) Ja, diese Wörter, die am Anfang standen, warum meinst du, kannst du dich besonders gut an die erinnern und wie die sich verhalten haben?
- (17) Ich glaube, weil man die am meisten gesehen hat. Also die waren fast immer dabei oder ich glaube, die waren sogar immer dabei und damit war das die häufigste Wiederholung. Der Rest hat sich glaube ich stärker verändert, als die am Anfang.

- (18) Okay, ja. Wenn du das Experiment jetzt nochmal machen würdest oder einen Test dazu machen würdest, wie gut denkst du, würdest du da abschneiden?
- (19) Ist ja schon eine Weile her, also wahrscheinlich müsste ich mich kurz wieder reinfinden, aber ich denke, wenn man sich es nochmal angeschaut hat, würde es wahrscheinlich relativ gut gehen.
- (20) Okay, ach schön. Wie schwer fandest du es denn im Vergleich dazu, die Syntaxregeln zu lernen für Python, für die Programmiersprache, die wir gelernt haben?
- (21) Eigentlich nicht schwierig, weil ich sie auch zum Teil schon kannte, weil ich ja nicht wusste, welche Sprache vorkommt und gerade bei Python kannte ich schon ein paar von den Regeln, aber die waren jetzt auch nicht sonderlich kompliziert, also es war eigentlich relativ leicht. Und weil die Regeln vorgegeben waren und nicht nur Beispiele, aus denen man sich die Regeln ableiten könnte, war es glaube ich sogar einfacher, das nachzuvollziehen als mit dieser künstlichen Sprache.
- (22) Ah, okay. Kannst du dich da an einzelne Aspekte erinnern aus der Programmiersprache, die für dich irgendwie aufgefallen sind?
- (23) Zum Beispiel die Zeileneinrückung oder so, die ist ja relativ typisch und das ist halt eine Regel, wo man es jetzt nicht nur zufällig herausgefunden hat, weil es immer wieder so geschrieben war, wo direkt gesagt wurde, nach einer if-Abfrage oder bei den Schleifen muss halt nach der Einleitung der Schleife oder nachdem eben das erste Statement kam, die weiteren Anweisungen eingerückt sein, bzw. muss das Statement wie das if-Statement oder so immer mit einem Doppelpunkt quasi abgeschlossen werden und danach kommt dann diese Einrückung. Also es ist so eine Syntaxregel, die sich glaube ich relativ gut merken lässt, weil es auch irgendwie logischer erklärbar ist.
- (24) Also ist dir das besonders aufgefallen oder im Gedächtnis geblieben, weil es so spezifisch war für die Programmiersprache?
- (25) Es ist glaube ich nicht in allen Programmiersprachen eine Regel, weil es ja zum Teil dann auch in Klammern eingefasst ist. Also ja, es lässt sich halt relativ gut, also ich meine es lässt sich sowieso immer machen, aber in der Sprache ist es halt eine Regel, dass es halt so sein muss, weil es ja keine Klammern gibt, die das Ganze einfassen und auch kein Semikolon am Ende jeder Anweisung, was das Ganze irgendwie abschließen würde und deswegen ist es eben aufgefallen, dass es halt in der Sprache so gemacht ist, dass durch diese Einrückung und durch das Formelle irgendwie so gestalten muss, dass es logisch erschließbar ist und dass eben auch der Computer dann das Ganze verarbeiten kann.
- (26) Ja, ja. Wenn du jetzt so darüber nachdenkst, über das Experiment und das Lernen der Syntax der Programmiersprache, sind dir persönlich bei dir jetzt irgendwelche Parallelen aufgefallen zwischen beiden?
- (27) Naja, es gibt halt bestimmte Regeln, die irgendwie eingehalten werden müssen, also zum Beispiel, was ich vorhin gesagt hatte, mit diesen zwei Wörtern, die immer irgendwie am Anfang stehen mussten. Es kann halt nicht einfach irgendein anderer Teil der Sprache am Anfang stehen, das heißt nur jetzt bei Pfeifen kann es nicht einfach, ich sag mal, so eine

Fallunterschaltung mit Ls anfangen, weil es halt mit F anfangen muss und das wäre, glaube ich, so eine Parallele.

- (28) Okay. Wir haben ja beim Experiment einmal kurz schon erzählt, dass wir so ein bisschen gucken wollen, ob diese künstliche Sprache oder das Lernen einer künstlichen Sprache generell ein Zwischenschritt sein kann zwischen nicht programmieren können und programmieren können. Hattest du das Gefühl, dass das ein logischer Zwischenschritt ist oder könntest du dir auch einen anderen Zwischenschritt vorstellen, der auch gut passen könnte, bevor man anfängt programmieren zu lernen?
- (29) Also es ist insofern ja logisch, wie ich es gerade schon erklärt habe, dass es gewisse Parallelen gibt, also denke mal, um grundsätzlich überhaupt in so ein Abstraktionsvermögen rein zu kommen. Es ist vielleicht nicht schlecht, wenn man sich vorher eine künstliche Sprache anschaut, die einfach kein Ziel oder so hat, also wo du jetzt nicht vor Augen hast, ich muss das und das in einem Programm umsetzen und wie kann ich das machen, sondern dass du erstmal grundsätzlich die Idee davon bekommst, irgendeine Grammatik zu lernen, die nicht direkt zu einer Sprache, mit der man sich unterhalten würde oder so dient, sondern die ein bisschen abstrakter ist und trotzdem die Regeln zu verstehen. Ich wüsste aber nicht, wie man das jetzt irgendwie ersetzen sollte oder was man sonst machen könnte.
- (30) Okay, alles klar. Gut, das hört sich schön an. Wir würden jetzt ein bisschen weitermachen mit Fragen zum Vorkurs, es sei denn, du hast noch Rückfragen zum Experiment.
- (31) Wie hilfreich war für dich diese Aufteilung in die Einführung in die grundlegende Syntax und Semantik von Python und in diese Templates, Plänevorlagen, wie auch immer man es nennen möchte?
- (32) Ich fand die Aufteilung gut, weil das habe ich bisher so noch nicht gesehen. Also wir hatten zum Beispiel auch in der Schule so Basic-Informatik mit so ein bisschen Programmieren, da war das aber nicht so gemacht oder da hatte man nicht das Gefühl, dass ein Konzept dahinter steht. Aber so die Templates fand ich eigentlich eine ganz gute Idee, weil du hast immer einen Abschnitt irgendwie gemacht, der halt etwas, ich sag mal, wenig greifbar in der Luft steht, wenn du halt nur sagst, okay, damit kannst du irgendwie eine Abfrage machen, aber wenn du das Ganze konkret umsetzt und dann halt sagst, okay, damit kannst du dann zum Beispiel Variablen vertauschen oder damit kannst du das und das machen und direkt eine Anwendung dazu hast, fand ich eigentlich eine gute Idee. Das hat das auf jeden Fall gut gefestigt, dann auch noch mal die Übung dazu zu machen und sich selber noch mal Gedanken zu machen, wie man sowas schreiben kann.
- (33) Dann dazu ergänzend, wie nützlich war die Aufteilung in Lesen und Schreiben für dich?
- (34) Also ich hatte es ja schon mal gesagt, da ich Python schon so ein bisschen kannte, fand ich es nicht so schwierig zu lesen, also das Ganze zu lesen, deswegen hätte ich glaube ich den Schritt vielleicht nicht unbedingt gebraucht, aber ich finde es grundsätzlich eine gute Idee, wenn man eine neue Sprache hat, die man noch nie gesehen hat, dass man sich das Ganze erst mal anschaut und überhaupt erst mal sieht, wie man eigentlich so

eine Programmiersprache auffasst, bevor man anfängt, das Ganze dann irgendwie selber umzusetzen und zu schreiben.

- (35) Wie war die Reihenfolge des Kurses für dich? Also gab es zum Beispiel, du konntest ja schon programmieren oder zumindest teilweise programmieren, aber gab es Konzepte, die für dich eher vor anderen hätten beigebracht werden sollten?
- (36) Ich denke nicht, also ich fand die Reihenfolge gut so, wie die war. Du konntest zwar schon programmieren, aber gab es für dich Konzepte oder Programmierkonstrukte, die besonders schwierig waren? Ich glaube nicht, dass es besonders schwierig war, so gerade bei den anfänglichen Sachen, also das war ja nur eine Grundidee davon, wie man jetzt zum Beispiel eine Schleife macht oder so. Es wäre vielleicht schwieriger geworden, wenn da mehr verschachtelt gewesen wäre, aber es war nicht so schwierig. Ich denke mal, zumindest was ich so mitbekommen habe bei den anderen, war die Schwierigkeit vielleicht auch gar nicht so im informatischen Verständnis oder im Verständnis des Programmiercodes, sondern vielleicht eher so mathematische Sachen, dass das dann zum Teil nicht bekannt war, wie Zwischenwerte sind oder was genau jetzt eigentlich berechnet wird, weil das vielleicht nicht so schnell erfasst wurde. Ich glaube, rein vom Programmiercode war es nicht wirklich schwierig.
- (37) Das wäre auch generell interessant, was auch für dich vielleicht, weil du konntest schon programmieren, aber findest du, oder erst mal im Kurs, war für dich die Problemlösung schwieriger oder der Ausdruck der Problemlösung in einer Programmiersprache?
- (38) Es ging eigentlich beides, weil es gut gestaffelt war. Weil man immer erstmal einen Plan schreiben musste, wie man so ein Programm macht, hat man sich glaube ich besser Gedanken darüber gemacht, wie man das Ganze macht und das dann in Code zu formulieren, war nicht schwierig. Also grundsätzlich ist es wahrscheinlich schwieriger, eine Idee zu haben, wie man das Problem grundsätzlich in einer Schrittfolge umsetzen kann. Aber es war ganz gut gelöst dadurch, dass du diesen Plan eben immer erst machen musstest und dann nicht zuerst angefangen hast, einen Code zu schreiben und dann nicht mehr wusstest, wie du weiterkommst.
- (39) Hast du sowas wie Pläne oder zum Beispiel in der Schule musst du dir ein Solo-Code schreiben oder würdest du das auch weiterhin nutzen?
- (40) Wir hatten Pläne, aber die waren immer sehr unübersichtlich, deswegen hat man die tendenziell eher gemieden, weil es die Sache nicht unbedingt vereinfacht hat. Aber so wie wir es jetzt gemacht haben mit einer Schrittfolge oder beziehungsweise mit diesen Diagrammen, die du manchmal auf den Seiten dazwischen hattest bei den Schleifen, war es eigentlich ganz gut verständlich und das könnte ich mir vorstellen, dass es zumindest, wenn es ein bisschen komplexere Sachen sind, gut ist, wenn man sich das mal vorher verbildlicht und nicht nur im Code so sieht. Würdest du das bei komplexeren Aufgaben auch noch anwenden, also dass du dir vorher quasi eine Schrittfolge machst? Ja, ich denke schon.

- (41) Vielleicht noch, wo du jetzt gerade die Diagramme ansprichst, hättest du es hilfreich gefunden, wenn ich so Flussdiagramme überall gemacht hätte oder zumindest bei den Sachen, wo es sich anbieten würde, bei bedingten Anweisungen zum Beispiel?
- (42) Bei bedingten Anweisungen hätte man es vielleicht noch mal machen können, aber ich glaube, bei dem Rest, also was alles vor Schleifen war, ist es nicht unbedingt so nötig gewesen, weil solange ein Programm, was sich Schritt für Schritt abläuft, sehe ich nicht wirklich einen Grund dafür, das jetzt irgendwie noch mal bildlich darzustellen, weil wenn es keine Verzweigungen oder Wiederholungen gibt, ist es glaube ich egal, ob man es einfach nur so untereinander schreibt oder ob man es noch mal in Kästchen Faktum untereinander schreibt. Hast du Erfahrung mit anderen Programmiersprachen? Immer nur Grundlagen, also wir hatten ja am Anfang dieses Blatt, wo wir ausfüllen mussten, wie wir uns mit bestimmten Sprachen auskennen. JavaScript und PHP kann ich so ein bisschen, weil wir es in der Schule hatten und C und Java habe ich mir mal angeschaut, habe ich dann hier wirklich die Zeit da jetzt viel mitzumachen. Also gerade von Java habe ich glaube ich noch gar keine so richtige Idee und C schon eher so ein bisschen, aber es ist alles, es sind alles wirklich nur Grundlagen. Ja, also Java und C sehen sehr sehr ähnlich aus. Kannst du oder würdest du auch aus deiner Erfahrung mit den anderen Programmiersprachen, war das für dich hilfreich, dass wir es in Python gemacht haben und zum Beispiel nicht in C? Also ich glaube C oder auch Java hätte den zeitlichen Rahmen wahrscheinlich noch mehr gesprengt, weil Python ja den Vorteil hat, dass du im Prinzip einfach losschreiben kannst, ohne dass du erstmal irgendwie für die Grundlagen, ohne dass du da schon auf irgendwelche Bibliotheken zugreifen musst. Also ich glaube die Syntax der Sprache an sich ist glaube ich erstmal komplexer bei anderen Sprachen und deswegen ist Python vielleicht insofern eine ganz gute Wahl, dass du halt genau diese Schrittfolge, was ich jetzt schon gesagt hatte, oder auch dann, ich sag mal was jetzt schon Schleifen und Abfragen oder so betrifft ist, glaube ich besser in Python zu machen, weil du einfach schon losschreiben kannst, ohne dass du erst Ewigkeiten nur die Syntax der Sprache an sich erklären musst.
- (43) Ja, das war auch die, also du hast jetzt die Kontention in deiner Sprache gut verstanden. Theoretisch, ihr müsst ja als erstes C lernen, aber ja, das ist glaube ich, da bin ich erstmal einen Tag die Rede nach mir erklärt. Andererseits ist halt dann die Frage, weil Python ist ja ein bisschen atypisch, was so Programmiersprachen betrifft, also zum Beispiel ist es nicht in Klammern eingefasst, wenn du irgendwo eine Abzweigung, also so eine Verzweigung hast oder eine Schleife und vielleicht ist es auch dann schwierig sich umzugewöhnen, aber ich glaube trotzdem, dass es eine ganz gute Idee ist, wenn du nicht so viel Zeit hast, erstmal mit einer Sprache anzufangen, die du direkt schreiben kannst.
- (44) Ja, also es ist auch in der Forschung genau die Abwägung macht, alleine auch, weil wir haben ja keine Typen und so was, macht es Sinn, Python als erstes zu nutzen oder nehmen wir lieber was schwierigeres und da gibt es noch keine wirkliche Antwort, was die klügere Lösung ist.

- (45) Dann weißt du, was meine Gruppe gemacht hat, während ihres Experiment gemacht hat?
- (46) Nein.
- (47) Die haben eine Einführung in Git bekommen, weißt du, was Git ist?
- (48) Ich glaube, ihr habt es gesagt, aber ich habe es noch nicht gemerkt. Einfach Versionsverwaltung, so ganz simpel runtergebrochen. Das heißt, wenn ich im Normalfall, ich habe ein Problem, ich speichere irgendwie dreißigmal meine verschiedenen temporären Lösungsansätze dazwischen, weiß ich nachher eventuell oder aller Wahrscheinlichkeit nach nicht, wo die Lösung Schritt XY irgendwie das erste Mal behandelt worden ist. Und in Git kann ich einfach, wenn ich alles klug hochlade, schreibe ich jedes Mal einen Kommentar dahin, ich habe dies und das geändert. Du kannst von jedem Schritt aus, so gesehen, wieder ein neues Projekt erstellen. Du hast so gesehen eine komplette Historie deiner Dateien. Hättest du das sinnvoll gefunden, das zu lernen? Oder wie sinnvoll hättest du das gefunden, da eine Einführung zu bekommen?
- (49) Also es ist wahrscheinlich nicht falsch, weil ich es mitbekommen habe. Es ist ja ein bisschen wie Spiel immer mal zwischendurch speichern, damit du, falls du irgendwas falsch machst, wieder auf den vorherigen Spielstand zurückgreifen kannst oder so. Also verschiedene Backups zu machen und die ordentlich zu verwalten, denke ich, ist auf jeden Fall praktisch.
- (50) Ja. Wie war der Pre-Test und der Post-Test bzw. beide Aufgaben im Post-Test für dich?
- (51) Den Pre-Test fand ich relativ leicht oder dann auch erst danach im Post-Test, also der selbe Teil, weil das ja wirklich nicht über Variablen und so hinaus ging, wenn ich mich richtig erinnere. Also ich glaube, das war ja nur so ein bisschen Variablen hin und her schieben, das war relativ simpel. Und der zweite Teil war von der Idee her halt ein bisschen komplexer, aber der ließ sich auch noch dem Fokus eigentlich ziemlich gut machen.
- (52) [...]
- (53) Also vielleicht dazu noch, gab es im Skript oder bei den Aufgaben Probleme außer die Sachen, die wir schon besprochen haben?
- (54) Eigentlich nicht, nur bei den Leseraufgaben, aber das haben wir dann meistens besprochen, war manchmal nicht ganz klar, welche von den Antwortmöglichkeiten jetzt richtig sind. Also wenn es bei Multiple Choice war, da war ich mir manchmal nicht sicher, ob verschiedene gepasst haben oder eigentlich gar keins so richtig, aber das hatten wir dann, glaube ich, dann jeweils besprochen.
- (55) Dann vielleicht noch zu den Variablen-Tabellen, würdest du so etwas noch nutzen, wenn ja, in welchem Kontext?
- (56) Ich glaube, Variablen-Tabelle sehe ich immer so ein bisschen als Tool für eine Fehlersuche, wenn du irgendwo keinen Syntaxfehler hast, den dir ein Debugger anzeigen würde, sondern halt einen logischen Fehler. Und halt um ein Stück Code an sich zu verfolgen, ich glaube, zum Schreiben wäre das mir ein bisschen zu aufwendig, weil ich glaube, wenn du einen Plan gemacht hast und das irgendwie ein logisches Konzept hat, dann müsste das ganz gut funktionieren. Aber wenn du einen Fehler suchst oder so, oder

irgendwie was nicht richtig verarbeitet wird, dann wäre das wahrscheinlich ganz praktisch.

- (57) Ja.
- (58) Gibt es ansonsten noch Anmerkungen von dir, vor allem zu dem Vorkurs, also zu dem Programmier-Teil, etwas, was dir gefehlt hat? Oder auch einfach einen Schwerpunkt, der dir gefehlt hat oder so etwas?
- (59) Eigentlich nicht. Also ich fand es ein bisschen schade, dass wir damit nicht ganz fertig geworden sind, aber es war jetzt nichts, was direkt gefehlt hätte.
- (60) Wie fandest du denn den Rahmen generell? Findest du, der hätte länger gehen können, also mehr Stunden am Tag?
- (61) Ja, an sich schon. Also 10 bis 14 Uhr 30 ist er jetzt nicht so sonderlich groß.
- (62) Beziehungsweise hatten wir auch eine sehr lange Pause dazwischen, dass ich glaube, ein bisschen länger hätte man es schon ziehen können.
- (63) kannst du dir vorstellen, dass du Strategien aus dem Vorkurs mit übernimmst in dein Studium?
- (64) Was genau weißt du über Strategien?
- (65) Zum Beispiel unterschiedliche Sachen, die wir gemacht haben, wie jetzt zum Beispiel die Variabeltabellen, aber auch die Aufteilung, dass du dir erstmal beilegst, okay, was mache ich eigentlich mit dem Code, bevor man den aufschreibt. Oder den vielleicht erst in natürlicher Sprache aufzuschreiben und dann runterzuschreiben. Also so ähnlich wie wir es im Vorkurs gemacht haben.
- (66) Also ich denke ja, die grundsätzliche Herangehensweise werde ich auf jeden Fall beibehalten, weil das bisher ja ganz gut funktioniert hat. Also sich Gedanken darüber zu machen, wie man Probleme lösen kann, mit welchem Code man das Ganze machen könnte und das erstmal so aufzuschreiben als Plan oder als Diagramm oder wie auch immer und dann das Ganze quasi nur noch zu übersetzen, denke ich, ist eine gute Herangehensweise. Und das mit den Variabeltabellen hatte ich ja gerade schon erzählt, also würde ich das wahrscheinlich nicht nebenbei irgendwie machen, weil man das zu aufwändig wäre. Also ich würde es halt entweder direkt in der Entwicklungsumgebung machen, ich weiß nicht, [NAME] hat das, glaube ich, erzählt, dass er ja das irgendwie ein Tool dafür hat, oder ich würde es halt dann nur zur Federsuche oder so machen, wenn irgendwo was nicht richtig verarbeitet wird.