

Programmieren lernen

Beitrag von „Antimon“ vom 3. Februar 2023 19:50

Meine lieben Mitforistinnen und Mitforistinnen, insbesondere die Informatik-Unterrichtenden

Ich erwähnte bereits, dass ich mich fürs kommende Herbstsemester an der Uni für die Facherweiterung Informatik einschreiben will. Ich kann wirklich nicht programmieren, würde es mir vor Studienbeginn aber schon gerne anschauen. Ich kann bzw konnte mal sowas wie LaTeX und Matlab irgendwie bedienen. Es gibt viele online Plattformen mit Kursen zum selber lernen. Was könnt ihr mir denn da empfehlen? Ich fand das hier z. B. ganz attraktiv: <https://edley.de/>
Taugt das was?

Danke schon mal für jedwede brauchbare Hinweise



Beitrag von „wieder_da“ vom 3. Februar 2023 20:28

Ich habe diese beiden Kurse einmal gemacht: <https://www.coursera.org/learn/python> und <https://www.coursera.org/learn/python-data>. So präsent ist mir das nach drei, vier Jahren nicht mehr, aber ich denke, dass man das schon empfehlen kann. Es gibt natürlich mittlerweile sehr viele MOOCs gerade in diesem Bereich.

Was in so einem Grundlagenkurs natürlich etwas fehlt, ist die praktische Anwendung. Man schreibt Programme, die die Primzahlen bis 1000 auflisten, die Wörter nach der Länge ordnen und dergleichen. Mir wurde einmal dieser Kurs empfohlen, wenn man wirklich gelegentlich Routineaufgaben am Computer durch selbst geschriebene Programme erleichtern will: <https://www.udemy.com/course/automate/>

Beitrag von „Seph“ vom 3. Februar 2023 21:04

Einen sehr überschaubaren Einstieg in Grundkonzepte der Programmierung kann man z.B. auch im Arbeiten mit Scratch erhalten, welches inzwischen zunehmend an Schulen verwendet wird. Insofern lohnt sich eine Einarbeitung für die spätere Unterrichtspraxis vermutlich ohnehin.

Beitrag von „yestoerty“ vom 3. Februar 2023 21:49

Neben Scratch sah <https://codecombat.com> auch vielversprechend aus. Das hat ein Kollege im Unterricht eingesetzt.

Ich hab damals Java mit "Java ist auch eine Insel" gelernt. Ist aber 15 Jahre her.

Beitrag von „wieder_da“ vom 3. Februar 2023 22:12

Für den Einstieg in Scratch gibt es auch einen Kurs, der sich an Lehrer richtet: <https://ocw.tudelft.nl/courses/scratc...mming-teachers/> Kann man machen, ist aber eher etwas für Primarstufe und Sek I.

Mit Scratch geht schon einiges, als Naturwissenschaftlerin in der Sek II wäre Calliope vielleicht interessanter ... führt aber m. E. beides etwas weg von der Frage nach einem eigenen Einstieg ins Programmieren. Ich würde mir schon mal eine Programmiersprache ansehen, und wenn es in der Summe nur für 20 Stunden ist, damit man dann hinterher in den bunten Blöcken von Scratch und Calliope die Konzepte wiederentdeckt, die man z. B. bei der Arbeit mit Python kennengelernt hat.

Beitrag von „Alasam“ vom 3. Februar 2023 22:26

Zitat von Seph

Einen sehr überschaubaren Einstieg in Grundkonzepte der Programmierung kann man z.B. auch im Arbeiten mit Scratch erhalten, welches inzwischen zunehmend an Schulen verwendet wird. Insofern lohnt sich eine Einarbeitung für die spätere Unterrichtspraxis vermutlich ohnehin.

Die Scratch-Erweiterung [Snap!](#) funktioniert prinzipiell ähnlich per Drag and Drop, die Programme sehen ähnlich aus, aber hier kann man deutlich mehr in die Tiefe gehen. So lassen sich alle Anforderungen ans Programmieren im Informatikabitur in Niedersachsen mittels Snap! abdecken.

Inzwischen gibt es auch noch SciSnap! mit Bibliotheken für z.B. Bildbearbeitung, Diagrammerstellung, Mathematik, Datenanalyse, Datenbanken, Graphen, Neuronale Netze.

Beitrag von „SteffdA“ vom 3. Februar 2023 22:47

Ich denke, die Grundstrukturen einer Programmiersprache zu verstehen ist der Punkt.

Beitrag von „Avantasia“ vom 4. Februar 2023 13:28

Schau mal hier bei den "Bundeswettbewerben Informatik", da sind ganz viele Kurse (und noch mehr Material) zusammengetragen worden:

<https://bwinf.de/programmierenlernen/#c7169>

Und das Algorithmen-Arbeitsheft ist auch nicht zu unterschätzen:

<https://bwinf.de/fileadmin/juge...Arbeitsheft.pdf>

À+

Beitrag von „Antimon“ vom 4. Februar 2023 19:30

Danke euch allen. Ich habe mir jetzt doch bei Edley einen C++ Kurs gekauft, werde mich aber je nach Zeit und Lust sicher auch mit Scratch & Co vergnügen. Ausserdem habe ich mir bei Conrad ein Arduino Bastelset bestellt, ich bin gerade motiviert 😊

Ich glaube das Schwierigste ist gerade die Anmeldung an der Uni. Ich habe heute nen halben Tag damit zugebracht Zeugnisse zu fotografieren und Dokumente hochladen. Meinen Lebenslauf musste ich nach 10 Jahren auch mal wieder aktualisieren, den wollen die aber eh auch auf dem Migrationsamt für die Einbürgerung. Jetzt fehlt mir noch ein aktuelles Foto, dann kann ich endlich auf den Knopf drücken. Ist schon speziell nach so langer Zeit wieder das ganze Gedöns von vorne. Und es ist so ganz anders als im ersten Leben. 1999 gab's noch keine online-Anmeldung, da sass ich mit meinen Papierli in Heidelberg auf dem Studentensekretariat

ganz analog auf dem Boden und musste warten, bis A wie Archäologie vor mir endlich fertig war 😊

Beitrag von „Der Germanist“ vom 4. Februar 2023 23:09

Wird denn gar nicht mehr BASIC unterrichtet? 🤖

Beitrag von „kleiner gruener frosch“ vom 4. Februar 2023 23:13

BASIC? Ich habe damals (vor 20 Jahren) noch COBOL und Assembler gelernt.

Mir ist noch die Assembler-Klausur im Kopf, bei der der Dozent bei der Rückgabe sagte: "Und xyz - Ihr Programm war sehr ... kreativ. Sie haben während des Ablaufs des Programms das Programm selbst geändert und damit das gewünschte Ergebnis erzeugt. Riskant, aber kreativ. Sie bekommen eine 1. Aber mache sie das nie wieder. 😊 "

Beitrag von „raindrop“ vom 5. Februar 2023 00:03

[Zitat von Antimon](#)

Ich habe mir jetzt doch bei Edley einen C++ Kurs gekauft

Du hast vorher noch nie programmiert? Dann ist C++ schon sehr ambitioniert. Wenn du Lust drauf hast mach es. Die Sprache ist auf jeden Fall gut strukturiert und geeignet um sich die Grundkonzepte zu erarbeiten. Einige tiefergehende Konzepte habe es aber auch in sich. Besser geeignet für einen Einstieg wären meiner Meinung nach Java oder Python.

Beitrag von „raindrop“ vom 5. Februar 2023 00:05

[Zitat von kleiner gruener frosch](#)

BASIC? Ich habe damals (vor 20 Jahren) noch COBOL und Assembler gelernt.

uns haben sie vor 30 Jahren mit LISP und Fortran in der Uni gequält 😊

Beitrag von „kleiner gruener frosch“ vom 5. Februar 2023 00:07

Ähm ... okay, du hast gewonnen. Wobei. Cobol ist ... Cobol. Oder anders ausgedrückt. Hätte es im Jahr 32 nach Christus schon Computer gegeben - sie hätten sie in COBOL programmiert. 😊

Beitrag von „Antimon“ vom 5. Februar 2023 09:50

[Zitat von raindrop](#)

Du hast vorher noch nie programmiert?

Nein, aber ich weiss grundsätzlich worum es geht. An der Uni fängt es auch mit Python und Java an. Noch habe ich keinen Stress, es geht um nichts, deswegen schaue ich es mir jetzt ja auch schon an.

Beitrag von „Antimon“ vom 1. März 2023 19:03

Also, falls es sonst noch jemanden interessiert: Der Kurs bei Edley ist sehr gut. Ich glaube, ganz ohne irgendein Vorwissen, insbesondere im Bereich Mathe und Logik, würde man wohl etwas weinen. Es geht schon wirklich sehr strukturiert drum, die Sprache zu verstehen, aber das passt mir mehr als ne Schildkröte mit dem Kopf nicken zu lassen 😊

Beitrag von „c. p. moritz“ vom 1. März 2023 19:26

Zitat von Der Germanist

Wird denn gar nicht mehr BASIC unterrichtet? 🤔

Ich habe grüne Bildschirme und 64kb-Computer mit Datasette vor Augen. Ich bin immer wieder begeistert, was sich seitdem in Hard- wie Software getan hat. Captain Picard würde staunen.

Beitrag von „CandyAndy“ vom 2. März 2023 15:46

Zitat von Antimon

Meine lieben Mitforistinnen und Mitforistinnen, insbesondere die Informatik-Unterrichtenden

Ich erwähnte bereits, dass ich mich fürs kommende Herbstsemester an der Uni für die Facherweiterung Informatik einschreiben will. Ich kann wirklich nicht programmieren, würde es mir vor Studienbeginn aber schon gerne anschauen. Ich kann bzw konnte mal sowas wie LaTeX und Matlab irgendwie bedienen. Es gibt viele online Plattformen mit Kursen zum selber lernen. Was könnt ihr mir denn da empfehlen? Ich fand das hier z. B. ganz attraktiv: <https://edley.de/> Taugt das was?

Danke schon mal für jedwede brauchbare Hinweise 🌸

Würde unbedingt erstmal genau abchecken, was gefordert wird. Kann sonst durchaus sein, dass man sinnlos viel zu viel lernt. Die meisten Erweiterungen sind eher tiefenentspannt. Bei Interesse natürlich ignorieren.

Beitrag von „Schmidt“ vom 2. März 2023 15:57

Zitat von CandyAndy

Würde unbedingt erstmal genau abchecken, was gefordert wird. Kann sonst durchaus sein, dass man sinnlos viel zu viel lernt. Die meisten Erweiterungen sind eher

tieferentspannt. Bei Interesse natürlich ignorieren.

Wie kann man "sinnlos zuviel" lernen?

Beitrag von „s3g4“ vom 2. März 2023 16:10

[Zitat von Schmidt](#)

Wie kann man "sinnlos zuviel" lernen?

Ja das ist echt lästig. Passiert mir auch manchmal, dann gehe ich schnell einen saufen um den Ballast wieder los zu werden.

Beitrag von „Antimon“ vom 2. März 2023 16:19

[Zitat von CandyAndy](#)

Würde unbedingt erstmal genau abchecken, was gefordert wird.

Du Fuchs. Da wär ich echt nicht von selbst drauf gekommen! 🧐👍

Beitrag von „Antimon“ vom 2. März 2023 22:42

[Zitat von c. p. moritz](#)

Ich habe grüne Bildschirme und 64kb-Computer mit Datasette vor Augen. Ich bin immer wieder begeistert, was sich seitdem in Hard- wie Software getan hat. Captain Picard würde staunen.

Das finde ich allerdings auch ultra cool. Ich habe es vor Unzeiten mal mit C probiert und es gab einen Grund, warum ich entnervt aufgegeben habe. Die IDE, mit der ich arbeite, ist ja praktisch

idotensicher. Ich bin begeistert, wie komfortabel das alles geworden ist. Irgendwie hat es mich immer gestört, dass ich nicht programmieren kann, aber dann blieb es doch nur beim Hardware zusammenstecken. Ehrlich gesagt geht es mir auch gar nicht so sehr um die Facherweiterung, ich habe jetzt einfach Lust, das zu machen. Das Lehrdiplom ist einfach eine bequeme Art sich irgendein Papierli dafür zu holen.

Beitrag von „Wolfgang Autenrieth“ vom 2. März 2023 23:27

[Zitat von Der Germanist](#)

Wird denn gar nicht mehr BASIC unterrichtet? 🤔

Scratch ist nicht so weit entfernt 😊

Beitrag von „Antimon“ vom 30. September 2024 13:43

Meine Lieben ...

Mir kam grade in den Sinn, dass ich diesen Thread mal gestartet hatte und dachte, ich hole ihn wieder hoch. Unterdessen habe ich 2 Semester hinter mir und kann etwas berichten, falls jemand Ähnliches plant. Ich hatte also die beiden Edley-Kurse für C++ und Python gemacht, was retrospektiv auch eine gute Entscheidung war. Insbesondere C++ schon mal gesehen zu haben, hat mir persönlich fürs Verständnis recht geholfen. Mit Python kommt man halt sehr schnell sehr weit ohne wirklich zu wissen, was Python eigentlich so macht. Ich bin im regulären Bachelorstudiengang Computer Science eingeschrieben, da ist das mit dem "nicht wissen, was es macht" keine rechte Option. Die Naturwissenschaftler*innen lernen in Basel ein Semester lang Scientific Computing in Python, die Informatiker*innen fangen mit Java an. Da die Java-Syntax dann doch recht ähnlich zu den C-Sprachen ist, sass ich da also wirklich nicht ganz unwissend rum 😊 Ich persönlich würde auch jemanden, der sich für die Facherweiterung interessiert, nicht Scratch & Co. empfehlen, sondern auf jeden Fall sich selbst mit "ernsthaften" Programmiersprachen hinzusetzen und ich würde den Weg grundsätzlich gar nicht empfehlen, wenn man nicht wenigstens rudimentäre Vorbildung in dem Bereich mitbringt, weil ... nächster Absatz.

Der Workload ist bei den Informatiker*innen in Basel echt verschärft und es wird verdammt viel Eigenständigkeit erwartet. Die Vorlesungen sind aber sehr gut, wenn man kontinuierlich dabei ist, lernt man sehr viel. Im 2. Semester hatten wir das Programmierprojekt, das "darf" ich aber nächstes Semester wiederholen. Das einzige, was für mich echt mühsam ist, sind die ständigen Gruppenarbeiten, ich schrieb darüber in einem anderen Thread. Alles, was man mich alleine machen lässt, funktioniert, Absprachen mit verpeilten 20jährigen sind mit meiner Berufstätigkeit eher schlecht vereinbar. Unterdessen ist aber im 1. Semester noch eine zweite Person aufgetaucht, die für die Facherweiterung eingeschrieben ist, wenn alles gut geht, kann ich das Projekt mit ihr zusammen machen. Einen grossen Teil davon habe ich ohnehin bereits selbst geschrieben. Wenn man nur Teilzeit studiert und keine nennenswerten Vorkenntnisse mitbringt, hat man da entweder verdammt Glück mit der Gruppe in der man landet, oder man stirbt. Ich kenne eine Person in meinem Studienjahrgang, die von der Gruppe einfach mitgezogen wurde, die ist aber jetzt gerade dabei, dahinzuscheiden weil sie halt viel zu wenig Zeit hat um von Grund auf alles zu lernen. Die Vollzeit-Studierenden werden ja mit nichts anderem zugeschmissen, manchmal ärgere ich mich, dass ich überhaupt noch nebenher arbeite.

Dieses Semester habe ich zwei Veranstaltungen aus dem Bereich der Technischen Informatik, die es so in Basel eigentlich gar nicht gibt, die Forschungsschwerpunkte liegen auf Artificial Intelligence und Distributed Systems. Den Verantwortlichen scheint aber aufgefallen zu sein, dass unsere Maturand*innen immer weniger Ahnung davon haben, wie so ein Computer eigentlich von Innen ausschaut und so gibt es nun in der 2. Runde eine Veranstaltung die sich "Informatiklabor" nennt. Die erste bewertete Übungsaufgabe hatten wir letzte Woche in Präsenz: Da sind so Teile auf dem Tisch, baue daraus einen Rechner zusammen. Ratet, wer als erstes fertig war 😁 Die Übungsgruppenleiter*innen haben zu Beginn immerhin noch gefragt, wer sowas schon mal gemacht hat, gerade etwa 1/4 streckt da überhaupt noch auf. Ich bin mir sicher, vor 15 - 20 Jahren wären das im Studiengang Computer Science noch alle gewesen, die selbst schon mal gebaut haben. Das war ja vor 20 Jahren bei uns Chemiker*innen noch Usus, selber stecken war halt viel günstiger als fertig kaufen. Für die zweite Veranstaltung in dem Bereich werde ich in der 2. Hälfte ein kleines Bastel-Projekt machen müssen, mit dem Dozenten habe ich bereits gesprochen, man lässt es mich mit reduziertem Workload alleine machen. Es wird darum gehen, einen Arduino oder Raspberry Pi anzusteuern, allerdings mit direktem Registerzugriff in C. Hier ist die Erwartungshaltung jetzt ganz klar "C lernste selber, ne".

Daneben habe ich noch Pattern Recognition und sterbe mit Linearer Algebra und Statistik. Es ist einfach echt lange her, aber die Übungsblätter zwingen einen ja dazu, sich in alles wieder einzulesen. Auch an dieser Stelle: Wer nur in Teilzeit studiert und im Bereich Mathe aus dem ersten Leben diese Art von Vorbildung nicht mitbringt, kann's komplett vergessen. Die Vollzeit-Studierenden haben im 3. Semester Statistik, Pattern Recognition belegen die meisten erst im 5. Semester. Für die Facherweiterung sind die Mathe-Vorlesungen eigentlich gar nicht gefordert, nur überlebt man ohne schlichtweg die Informatik gar nicht. Das Tolle ist, ich verstehe in der Mathe gerade bei so einigem überhaupt erst, wofür das alles gut sein soll und

stelle fest, wie wenig Chemie dann am Ende eben doch nur mit Mathe zu tun hat. Irgendwann hat man uns den ganzen Kram mal reingestopft, aber auch an der Uni weitestgehend ohne Bezug zum Hauptfach. Ob das heute so viel anders ist, weiss ich nicht. Molecular Modelling hat sicher ein höheres Gewicht bekommen als vor 20 Jahren, aber fürs Synthesepraktikum braucht auch heute kein Mensch Lineare Algebra.

Das Fazit ist also: Das Studium ist für mich eine echte Bereicherung, auch wenn es verdammt anstrengend ist. Empfehlen würde ich es nur bedingt, man sollte wirklich nicht ganz ahnungslos in dem Bereich sein. Vollzeit geht es sicher auch ohne nennenswerte Vorbildung, aber dann beschäftigt man sich halt eben auch nur noch mit Informatik.

Edit: Diesen YouTube-Kanal wollte ich noch empfehlen, falls er den anwesenden Informatiker*innen nicht ohnehin schon bekannt ist:

[BroCode](#)

Der ist auch super für Leute, die "einfach nur so" Programmieren lernen wollen. Wirklich verstanden habe ich's aber erst mit den Übungsaufgaben an der Uni. Ich finde insbesondere Objektorientierung in solchen online-Tutorials oft echt schräg erklärt.

Beitrag von „Philio“ vom 30. September 2024 18:49

Danke für das Update ☐ Auch wenn ich früher vieles in diesem Bereich gemacht und auch eine Zeit lang in der Softwareentwicklung gearbeitet habe (und auch heute noch für mich persönlich programmiere), bestärkt mich dein Beitrag darin, nichts in diese Richtung zu machen ☐ Insgesamt hat das Informatikstudium für mich zu viele Aspekte und Inhalte, die ich nicht so reizvoll finde. Noch weniger reizvoll finde ich persönlich Informatik als Unterrichtsfach.

Dir drücke ich fest die Daumen, dass alles so aufgeht, wie du dir das vorgestellt hast ☐

Ah, aber einen Link zum Teilen an alle habe ich auch noch ☐

Falls ihr sowieso schon die Infrastruktur von GitHub kostenfrei nutzt oder diese nutzen möchtet, es gibt ein Education Angebot mit dem ihr auch sonst kostenpflichtige Angebote wie den GitHub Copilot oder Codespaces kostenfrei nutzen könnt. Habe mich dort mit meiner Schulmail Adresse registriert und musste ein Anstellungsdokument hochladen. Freigeschalten wurde ich ein paar Tage später.

Hier der Link:

<https://docs.github.com/en/education/e...on-as-a-teacher>

Beitrag von „Antimon“ vom 30. September 2024 19:44

Zitat von Philio

Noch weniger reizvoll finde ich persönlich Informatik als Unterrichtsfach

Ich habe ehrlich Angst davor. Ich glaube, im Moment polarisiert fast kein Fach so sehr wie Informatik. Noch unbeliebter dürfte nur Französisch sein 😊 Ich bin mir auch immer noch nicht sicher, ob ich es gut finde, dass Informatik Grundlagenfach und entsprechend aufdotiert wird. Für mich selber finde ich es gerade toll, so viel sinnvollen Bezug zur Mathe zu sehen. Aber die Vorstellung, das den Jugendlichen vermittelt zu können, ist vermutlich hoffnungslos romantisch.

Beitrag von „Volker_D“ vom 30. September 2024 20:08

Ich habe damals noch programmieren nur durch Selbststudium von Büchern gelernt. Geht auch, dauert aber etwas.

Wer nur mal so ins Programmieren einsteigen will, dem würde ich evtl. eher einen Kurs bei OpenHPI (Hasso Plattner Institut. Einer der 3 Gründer von SAP.) empfehlen. Da gibt es auch Kurse, die Schüler aus Jahrgang ~8 gut schaffen können.

Die haben aber auch andere tolle Themen, nicht nur programmieren oder Informatik.

Vorteil: Es ist nicht nur ein kostenloser Videokurs, sondern es gibt auch Aufgaben, die bewertet werden. Von Multiple Choice Aufgaben bis zu Programmieraufgaben. Die Kurse kann man immer machen, aber wenn man sie in den angegebenen Zeiträumen macht, dann sind dort auch viele Mitlernende, die man im Forum fragen kann und man kann eine Prüfung machen und bekommt auch noch ein Zeugnis. Einige Kurse kann man sich sogar fürs Studium anrechnen lassen.

Aktuell starten in Kürze:

Python – schnell und intensiv Programmieren lernen

und

Positive Lernbeziehungen gestalten - Einführung in das intus³ Beziehungslernen nach Dr. Helga Breuninger und Professor Dr. Wilfried Schley

siehe:

<https://open.hpi.de/>

Beitrag von „Wolfgang Autenrieth“ vom 1. Oktober 2024 00:12

[Zitat von Antimon](#)

Die erste bewertete Übungsaufgabe hatten wir letzte Woche in Präsenz: Da sind so Teile auf dem Tisch, baue daraus einen Rechner zusammen.

Das ist eine schöne Aufgabe für eine AG oder Technikgruppe. Meine Kids der Klasse 9 haben mehrere Schrobots (Schrottroboter) gebastelt.

BTW: Über das Diskettenlaufwerk wird ein Basic-Programm gestartet, das über OUT-Befehle ein 8-Kanal-Relais-Board die Stromversorgung der Birnchen und Motoren, sowie der (offenen) Festplatten ansteuert. Ein Schrobot hatte es dann auf die Dezemberseite des Kalenders der SCHULKUNST Baden-Württemberg geschafft 😊

[robot1.jpg](#)

Beitrag von „Philio“ vom 1. Oktober 2024 13:04

[Zitat von Antimon](#)

Ich glaube, im Moment polarisiert fast kein Fach so sehr wie Informatik.

Hm, ich gehöre nicht zu den Kritikern, ganz im Gegenteil - Informatik als Fach halte ich grundsätzlich für eine gute Idee. Ich würde es nur nicht gerne unterrichten wollen ☹️

Die konkrete Umsetzung stelle ich mir aber schwierig vor, besonders wenn nicht viele Unterrichtslektionen zur Verfügung stehen.

Deshalb habe ich tatsächlich mal einen Blick in euren aktuellen Lehrplan geworfen, den du andernorts erwähnt hattest - für mich sieht der nach einem ziemlichen Balanceakt aus, möglichst viele Perspektiven unterzubringen: Grundlagen, Technik, Programmieren, Gesellschaft. Andererseits wird mir aus dem Dokument heraus nicht ganz klar, welches Niveau erreicht werden soll ... nach relativ grundlegenden Konzepten sehe ich oft meiner Meinung

nach eher grössere Sprünge in den Anforderungen mit Begriffen wie SQL, XML und JSON, Dijkstra, logische Schaltungen etc. Und das alles in der ersten Klasse mit 2 Lektionen und einer Praktikumslektion (wenn ich den Plan richtig verstanden habe). Für mich als absolut Unwissenden, was Informatikunterricht angeht, sieht das ziemlich sportlich aus ...

Beitrag von „Antimon“ vom 1. Oktober 2024 13:21

Im Moment ist das so überhaupt nicht umsetzbar und im Grunde unterrichtet jede und jeder nach Belieben. Da Informatik mit der WEGM Grundlagenfach wird, muss per Gesetz eine Mindestanzahl an Jahreslektionen unterrichtet werden. Ich nehme an, dass sich die Zahl im Baselland verdoppeln wird, dann kann man schon ein bisschen was machen.

So wie ich das bis anhin erlebe, unterrichten die meisten Lehrpersonen ein bisschen Programmieren in Python und dann vor allem Datensicherheit. Unser einzig studierter Informatiker legt den Schwerpunkt auf Webdesign, also HTML/CSS/Javascript, Datensicherheit und macht noch irgendwas zur Maschinenintelligenz. Letzteres ist einfach der Tatsache geschuldet, dass er ein Abkömmling einer entsprechenden Arbeitsgruppe an der Uni Basel ist.

Mein Eindruck von der Informatik ist bisher, dass sie ähnlich erschlagend facettenreich wie die Biologie ist. Chemie ist wirklich gemütlich dagegen, da stellt sich kaum die Frage, was man eigentlich unterrichten soll. Ich hätte wahrscheinlich eher Spass an Technischer Informatik, die Affinität scheint unter meinen Mitstudierenden allerdings nicht sehr weit verbreitet zu sein. Kaum irgendjemand belegt da Physik im Nebenfach, die meisten nehmen sowas wie BWL.

Beitrag von „Philio“ vom 2. Oktober 2024 14:12

Dein Vergleich zu Bio ist ziemlich gut, denke ich. Würde ich auch nicht unterrichten wollen ☹ Da hätte ich schon im Studium keine Chance gehabt - bin kein Auswendiglerner und diese Bücher mit 1000+ Seiten, wie Stryer oder Lodish, pro Semester reinzuziehen, wäre für mich unmöglich gewesen - von wegen Bio ist einfacher als Physik ☹ Für mich ist Bio eines der schwierigsten Fächer, die ich mir vorstellen kann.

Technische Informatik, hmm... ja, in dieser Richtung waren tatsächlich meine Aufgaben in der Industrie, hauptsächlich Embedded Systems. Dazu muss ich zugeben, dass mich die technischen Hintergründe mal mehr und mal weniger interessiert haben. Hinderniserkennung für autonomes Fahren fand ich spannend (ja, das war damals vor über 10 Jahren schon „in“ ☹),

die Steuerung von Motoren eher weniger. Bin halt doch Theoretiker ☹️ mit technischen Umsetzungen kann man mich eher nicht ködern.

Die Studierenden kann ich deshalb gut verstehen - für Webentwicklung o.ä. muss man nicht noch die Funktionsweise von anderen technischen Systemen verstehen. BWL ist als Kombination sehr clever, wenn man an den Arbeitsmarkt denkt - Hintergrundwissen in diesem Bereich ist für viele der attraktiven Jobs wahrscheinlich nützlicher als technisches Wissen.

Beitrag von „Antimon“ vom 6. Oktober 2024 13:16

[Zitat von Philio](#)

BWL ist als Kombination sehr clever, wenn man an den Arbeitsmarkt denkt

Das auf jeden Fall, ich hatte auch in der Chemie viele Mitstudierende, die im Hauptstudium sowas als Nebenfach belegt haben. Mich kannst du jagen mit allem, was ins Kaufmännische geht, ich bin heilfroh, dass sich genügend andere Leute dafür interessieren. Dass das nützlich ist, verstehe ich absolut nur ich kann mich null und nicht dafür begeistern.

Theoretikerin bin ich eben auch nicht. Ich kann wohl nachvollziehen, dass Menschen sich für Mathe begeistern, ich hätte das aber nicht studieren können. Rein intellektuell hätte ich es wohl irgendwie hingewurstet, mich dabei aber irgendwann zu Tode gelangweilt. Ich bin im letzten Semester in der Theoretischen Informatik auch fast eingegangen. Beim Lernen für die Prüfung fiel mir dann auf, ach, so schwierig ist es auch wieder nicht, aber währenddessen war es für mich ein einziges ... häh ... wieso muss ich das jetzt?! Wenn es mich nicht zugleich aus dem Programmierprojekt rausgeschmissen hätte, wär's wahrscheinlich besser gegangen. Ich muss einfach mit den Händen was zu tun haben, dann kann ich nebenher auch den Theoriekram ganz gut laufen lassen 😊

Beitrag von „Philio“ vom 7. Oktober 2024 16:53

[Zitat von Antimon](#)

Mich kannst du jagen mit allem, was ins Kaufmännische geht, ich bin heilfroh, dass sich genügend andere Leute dafür interessieren. Dass das nützlich ist, verstehe ich absolut

nur ich kann mich null und nicht dafür begeistern.

Als jemand, der mal eine kaufmännische Lehre gemacht hat, kann ich dich da

sehr gut verstehen ☹️ Wenn ich mich bei meinen Lernenden vorstelle, dann sage ich immer, ich hätte die Lehre gemacht, die alle diejenigen machen, die keine Ahnung haben, was sie sonst machen sollen. Bisher war es noch jedes Mal so, dass dann sofort jemand „KV!“ gerufen hat - auch und gerade in Klassen, in denen viele diese Lehre selber gemacht haben ☹️☹️☹️

Zugegeben, BWL und Rechnungswesen finde ich nicht so aufregend... aber VWL finde ich richtig spannend, speziell alles, was in Richtung Finanzmärkte geht. Auch alles rund um „Personal Finance“.

Daraus lassen sich auch spannende Programmierprojekte machen, um mal wieder die Kurve zum eigentlichen Thema zu kriegen ☹️ Zum Beispiel automatisierte Analyse von Kursen, Indices oder Unternehmensdaten mit Live-Daten aus dem Web und Visualisierung als Dashboard mit Dash oder Streamlit ☹️

Beitrag von „Antimon“ vom 3. Januar 2025 05:13

Also ... Ich getraue mich mal schon vor dem offiziellen Ende des Semesters zu verkünden, dass ich eigentlich nicht mehr verkacken kann. Im Pattern Recognition ist schon die Zwischenprüfung bestanden und die Übungen mit voller Punktzahl abgeschlossen, in der Hauptprüfung im Januar könnte ich mir eine 2 leisten um insgesamt noch auf eine 4 zu kommen. Das schaffe ich nicht, glaube ich, das grenzt ja an Totalverweigerung. Ansonsten kämpfe ich gerade mit meinem Arduino: Wie stampft man möglichst viel Funktionalität auf möglichst wenig Speicherplatz klein. Die schriftliche Prüfung in Computer Architecture ist schon geschrieben und sicher bestanden, fürs Projekt will ich mal mindestens eine 5.5 haben. Ich werde anspruchsvoll 😊 (Und ich weine, wenn was anderes rauskommt ...). Ach so ... Und um zur Ausgangsfrage zurückzukommen... Ich habe sowas wie eine "Lieblingssprache" gefunden: C++. Ich befürchte nur, das kann man jungen Menschen an der Schule nicht antun.

Beitrag von „SteffdA“ vom 3. Januar 2025 09:33

[Zitat von Antimon](#)

Ich habe sowas wie eine "Lieblingssprache" gefunden: C++. Ich befürchte nur, das kann man jungen Menschen an der Schule nicht antun.

Es gibt Steigerungen. Ich denke da an Forth, Prolog oder Lisp.

Ansonsten: Glückwunsch!

Beitrag von „Volker_D“ vom 3. Januar 2025 10:48

[Zitat von SteffdA](#)

Es gibt Steigerungen. Ich denke da an Forth, Prolog oder Lisp.

Dann hätte ich eher Rust als Steigerung genannt.

c++ ist definitiv meine Lieblingssprache. In der Sek. I komme ich aber im Grunde nie bis zur Objektorientierung. Evtl. mache ich die "Basics" einfach zu ausführlich? Ich habe dieses Jahr mal wieder vor mit Arduinos zu arbeiten. Hat den schönen motivierenden Teil Technik/Physik gleich mit dabei. Macht es aber auch wieder schwieriger, weil die Schüler auch mit der Physik "kämpfen" müssen. Mal gucken wie es dieses Jahr laufen wird.

PS: Beim ersten Satz habe ich einen kurzen Moment überlegt was genau gemeint ist. a) Es steht noch eine Leistung aus und egal was ich mache, ich habe es geschafft. Oder b) Es sind alle Leistungen abgegeben worden und alle waren unzureichend.

Beitrag von „puntino“ vom 3. Januar 2025 11:49

Neuen Code würde ich nicht in C++ schreiben. Nicht ganz kleine Klitschen wie Google verabschieden sich davon, weil die Speichersicherheit nicht garantiert werden kann.

Zitat

- We see no realistic path for an evolution of C++ into a language with rigorous memory safety guarantees that include temporal

safety.

[...]

We consider it important to complement a transition to memory safe languages for new code and particularly at-risk components with safety improvements for existing C++ code, to the extent practicable.

[Quelle: Google Blog](#)

Didaktisch halte ich C++ aber auch für Schüler für eher ungeeignet. In dem Zusammenhang sind besonders Pointer inklusive deren Dereferenzierung für Neueinsteiger eher im Bereich "Raketenwissenschaften" anzusiedeln.

Beitrag von „kleiner gruener frosch“ vom 3. Januar 2025 11:59

Es geht doch nichts über Cobol

Alternativ (aber ganz andere Schiene): Assembler.



Beitrag von „Volker_D“ vom 3. Januar 2025 12:16

[Zitat von puntino](#)

Neuen Code würde ich nicht in C++ schreiben. Nicht ganz kleine Klitschen wie Google verabschieden sich davon, weil die Speichersicherheit nicht garantiert werden kann.

[Quelle: Google Blog](#)

Didaktisch halte ich C++ aber auch für Schüler für eher ungeeignet. In dem Zusammenhang sind besonders Pointer inklusive deren Dereferenzierung für Neueinsteiger eher im Bereich "Raketenwissenschaften" anzusiedeln.

Warum zitiert du nicht vollständig?

"A large-scale rewrite of all existing C++ code into a different, memory-safe language appears very difficult and will likely remain impractical."

Es gibt eben Bereiche, da gibt es kaum eine Alternative. Zum Beispiel bei den Programmen, die ich schreibe. Da kommt es in erster Linie auf Geschwindigkeit an und in zweiter Linie brauche ich ein GUI.

Im Bereich Speichersicherheit und Geschwindigkeit würde sich für mich zum Beispiel Rust anbieten. Praktisch nutzen kann ich es aber nicht, weil es für Rust noch keine guten Bibliotheken gibt, mit denen man ein GUI schreiben kann. Es gibt zwar andere Sprachen, die das bieten, dort mangelt es dann aber oft an Geschwindigkeit, Speichersicherheit oder die Programme lassen sich nicht auf den "großen" Betriebssystemen kompilieren.

Ich würde darauf tippen, dass in den nächsten 10 bis 20 Jahren evtl. Rust erstmal die meisten C Programme ersetzen wird. Und dann, evtl. in 30 bis 40 Jahren die C++ Programme. Das wird dauern. C/C++ ist und bleibt gute Grundlage für die nächsten Jahre.

Beitrag von „TroyMcLure“ vom 3. Januar 2025 12:16

[Zitat von puntino](#)

Neuen Code würde ich nicht in C++ schreiben. Nicht ganz kleine Klitschen wie Google verabschieden sich davon, weil die Speichersicherheit nicht garantiert werden kann.

Ja, aber man muss da schon zwischen Programmieren und Softwareentwicklung unterscheiden. Ich glaub', dass die Schüler keine sicherheitsrelevanten Anwendungen entwickeln.

Beitrag von „Wolfgang Autenrieth“ vom 3. Januar 2025 12:27

[Zitat von kleiner gruener frosch](#)

Es geht doch nichts über Cobol

Alternativ (aber ganz andere Schiene): Assembler.



Assembler ist cool. In den 70er Jahren hatte mein Kumpel einen Cosmos-Baukasten mit programmierbarem NIC zu Weihnachten bekommen. Den haben wir dann gemeinsam in Maschinensprache programmiert.

Er studierte nach dem Abi Informatik in Berlin, ich hab' Kunst und Mathe auf Lehramt studiert.

Er war dann an der Entwicklung von StarOffice (dem Vorläufer von LibreOffice) beteiligt und hatte eine eigene Firma, die er irgendwann Anfang der 90er verkauft hat. Damals hat er reihum gefragt, wo es auf der Welt nett sei, um sich niederzulassen, falls genug Geld vorhanden sei.

Seitdem lebt der Sack auf Maui (Hawaii) 🤪

Beitrag von „Antimon“ vom 3. Januar 2025 12:31

Dave Plummer hat letztens mal Zig in einem seiner Videos vorgestellt. Sieht tatsächlich nach einer Alternative zu C und C++ im Bereich Systemprogrammierung aus.

Zitat von puntino

In dem Zusammenhang sind besonders Pointer inklusive deren Dereferenzierung für Neueinsteiger eher im Bereich "Raketenwissenschaften" anzusiedeln

Naja, bei C++ finde ich gerade den Teil jetzt erheblich weniger wüst als in C. Und auch in C ist es schlussendlich nur eine mühsame Tipperei und man muss halt dran denken, dass es für jedes malloc irgendwo ein free geben muss. Wirklich gecheckt, was ein Referenzdatentyp sein soll, wie verkettete Listen funktionieren, etc. habe ich dieses Semester, als wir eben angefangen haben in C und C++ zu programmieren. 1. Semester war Java, das war viel einfach irgendwas tippen, von dem man nur erahnt, warum eigentlich. Und Python führt sowieso irgendein seltsames Eigenleben, das ist mir hochgradig unsympathisch.

Zitat von kleiner gruener frosch

Alternativ (aber ganz andere Schiene): Assembler

Man lernt in Basel nur noch, wie es prinzipiell geht und quält sich mal durch einen Simulator. So habe ich aber wahrhaftig final erst verstanden, wie Rekursion funktioniert. Es hat schon noch seine Daseinsberechtigung, aber dass man es noch wirklich lernt ist glaube wenig sinnvoll.

Beitrag von „Volker_D“ vom 3. Januar 2025 12:31

Ein Irrglaube ist ja, dass man in C++ mit "Pointer inklusive deren Dereferenzierung" arbeiten muss.

Man kann komplette c++ GUI Anwendungen schreiben, ohne damit überhaupt direkt in Berührung zu kommen und ohne das zwingend verstehen zu müssen.

Beitrag von „Volker_D“ vom 3. Januar 2025 12:36

Bei Python ist ja der Witz, dass ein sehr großer Teil von Python in c geschrieben ist, weil Python in Python geschrieben einfach zu langsam ist.

Beitrag von „TroyMcLure“ vom 3. Januar 2025 12:43

Habt ihr eigentlich auch Erfahrungen in der Softwareentwicklung, insbesondere in Bereichen wie Design Pattern und UML? Ich frag' deshalb, weil mir zu diesen Themen regelmäßig der Austausch fehlt. Treib' mich dann regelmäßig bei Stackoverflow rum, wobei die Community ja nicht so verständnisvoll auf didaktische Fragen reagiert.

Beitrag von „Antimon“ vom 3. Januar 2025 12:51

 [Zitat von Volker_D](#)

Ein Irrglaube ist ja, dass man in C++ mit "Pointer inklusive deren Dereferenzierung" arbeiten muss.

Man kann komplette c++ GUI Anwendungen schreiben, ohne damit überhaupt direkt in Berührung zu kommen und ohne das zwingend verstehen zu müssen.

Eben das. Man kann ja einfach dynamisch mit Objekten arbeiten, wenn man keine Lust auf Zeiger hat und die Aufganstellung nicht zwingend kritisches Speichermanagement verlangt. Grade dass es so und eben auch anders geht, finde ich bei C++ sehr sympathisch. Und wie du schreibst, Qt z. B. ist für GUI Programmierung nun auch nicht schlechter als der Scene Builder für Java. Man kommt mit C++ irgendwie überall hin.

Zitat von Volker D

Bei Python ist ja der Witz, dass ein sehr großer Teil von Python in c geschrieben ist, weil Python in Python geschrieben einfach zu langsam ist.

Python ist für Leute, die eigentlich keine Lust auf Informatik haben. Es hat schon einen Grund, dass es im Scientific Computing unterdessen so weit verbreitet ist. Als Naturwissenschaftlerin interessiere ich mich nicht für Code-Optimierung sondern für die Auswertung meiner Daten. Wobei in dem Bereich Fortran wohl auch nie aussterben wird 😊

Beitrag von „Volker_D“ vom 3. Januar 2025 13:20

Zitat von TroyMcLure

Habt ihr eigentlich auch Erfahrungen in der Softwareentwicklung, insbesondere in Bereichen wie Design Pattern und UML?

Zu "Design Pattern" und "UML": Im Grunde nur das, was so auch in Fachbüchern steht. Aber "richtig" genutzt: nein.

Zu "Softwareentwicklung". Ja. Ich programmiere Software, die wirklich im Berufsleben von anderen Leuten produktiv angewendet wird. Also nicht so ein Hobby "Ich programmiere mal Taschenrechner oder Sudoku".

Schwerpunkt bei mir C++/Qt. GUI, welche NP Probleme lösen kann. Daten werden u.a. in einer SQL Datenbank gespeichert. Daten werden aus verschiedensten Formaten gelesen (xml, csv, ...) und in verschiedensten Formaten exportiert (html, cvs, ...). (Da ist sogar etwas css, JavaScript und LaTeX dabei. Aber da bin ich nur auf Anfängerniveau). Ich kompiliere die Software für Windows, MacOS, Linux, x86, ARM. WebAssembly habe ich mal ausprobiert, geht aber noch nicht gut genug. Versuche ich mal in ein paar Jahren nochmal.

Eine gute Anlaufstelle für Design Pattern und UML kann ich nicht nennen. Für Qt ist definitiv das offizielle Qt Forum die erste Anlaufstelle bei Fragen (Nach der sehr guten Dokumentation, die (fast) alle Qt Dinge sehr gut vorstellt.).

Beitrag von „Philio“ vom 3. Januar 2025 17:24

[Zitat von Antimon](#)

Python ist für Leute, die eigentlich keine Lust auf Informatik haben. Es hat schon einen Grund, dass es im Scientific Computing unterdessen so weit verbreitet ist. Als Naturwissenschaftlerin interessiere ich mich nicht für Code-Optimierung sondern für die Auswertung meiner Daten. Wobei in dem Bereich Fortran wohl auch nie aussterben wird

Also ich programmiere mega gerne in Python ☐☐

In C++ würde ich nichts mehr machen wollen, dazu ist mir die Sprache bzw. das ganze Ökosystem inzwischen zu Old School und meine Liebessprache war C++ noch nie, auch nicht, als ich vor einer gefühlten Ewigkeit (wobei... nicht nur gefühlt, ist schon fast 10 Jahre her ☐☐) damit noch meine Brötchen verdient habe ☐☐

Wenn eine systemnahe Sprache, dann würde ich mir tatsächlich Rust überlegen. Oder abwarten, ob Google mit Carbon tatsächlich den grossen Wurf schafft.

Fortran mochte ich immer schon gerne, da werde ich vielleicht mal wieder einen Blick einwerfen, wenn ich Zeit dazu habe.

Beitrag von „Antimon“ vom 3. Januar 2025 17:32

[Zitat von Philio](#)

Also ich programmiere mega gerne in Python ☐☐

Du bist eindeutig komisch ... 😄

(Und ich habe eigentlich eh keine Ahnung sondern klugscheisse nur so vor mich hin. Nee, im Ernst ... Allein die dynamische Typisierung macht mich bei Python schon nervös. Da funktioniert wahrscheinlich mein Hirn einfach total old school.)

Beitrag von „Volker_D“ vom 3. Januar 2025 18:15

Wobei "c++ ist old school" so auch nicht stimmt. C++ entwickelt sich immer noch sehr aktiv weiter. Datentypen müssen nicht mehr zwingend vorgegeben werden. (Stichwort "auto". Sinn und Einsatzbereich ist aber ganz anders als bei Python. Man kann jetzt auch lambda-Ausdrücke schreiben, ...). Ich gucke mir auf YouTube schon mal ganz gerne die Vorträge c++ Compilerentwicklern an. Es ist schon interessant zu sehen in wie unterschiedliche Richtungen sie die Sprache weiterentwickeln möchten und warum sie einige "modernere" Techniken nicht aufnehmen wollen oder können bzw. warum sie schon mehrfach einige "moderne" Features aus aktuellen Versionen wieder entfernt haben.

Beitrag von „Philio“ vom 3. Januar 2025 19:41

[Zitat von Antimon](#)

Du bist eindeutig komisch ...

Ich weiss ☐☐ Du bist auch nicht die erste, die das feststellt... ☐☐

Das eigentlich Schicke an Python finde ich persönlich die Libraries und damit verbunden die Tatsachen, dass mit einem Paketmanager wie Conda (ich nutze conda-forge) das Auflösen der Abhängigkeiten in den meisten Fällen ziemlich gut klappt - zumindest, soweit es mich betrifft.

Im Gegensatz zu Julia, die ich von der Idee her super finde, die aber praktisch unter einem ziemlichen Entwicklungschaos zu leiden scheint - die Community hat sich da anscheinend noch nicht auf einen gemeinsamen roten Faden einigen können. Jedenfalls war das so, als ich vor

einem halben Jahr mal etwas probieren wollte und an inkompatiblen Paketen gescheitert bin.

Beitrag von „raindrop“ vom 3. Januar 2025 21:30

Zitat von Volker D

Ein Irrglaube ist ja, dass man in C++ mit "Pointer inklusive deren Dereferenzierung" arbeiten muss.

Man kann komplette c++ GUI Anwendungen schreiben, ohne damit überhaupt direkt in Berührung zu kommen und ohne das zwingend verstehen zu müssen.

exakt so. Obwohl es für die Einführung der Grundkonzepte der Programmierung in der Schule einfachere Programmiersprachen gibt als C++.

Es kommt bei den Programmiersprachen aber doch auch immer auf den Einsatzzweck an.

Ich programmiere gerne in C++, aber wenn ich was mit neuronalen Netzen oder KI programmieren möchte, kommt man momentan an Python,

aufgrund der umfangreichen Bibliotheken zu dem Thema, nicht vorbei.

Aber natürlich hat sich diese Sprache in der Wissenschaft durchgesetzt, da die Syntax sehr gummiartig benutzerfreundlich ist...

Beitrag von „Volker_D“ vom 3. Januar 2025 22:18

Ja, KI (bzw. das nutzen der KI-Bibliotheken; die Bibliotheken selbst sind oft in C geschrieben) und grafische Aufbereitung von Daten in der Wissenschaft z.B. mit Mathplotlib sind Bereiche, in denen Python bevorzugt genutzt wird.

Ich werfe Schüler ja nicht (mehr) ins kalte Wasser. In den letzten Jahren habe ich immer zuerst Scratch gemacht.

Beitrag von „Meer“ vom 3. Januar 2025 23:42

Zitat von TroyMcLure

Habt ihr eigentlich auch Erfahrungen in der Softwareentwicklung, insbesondere in Bereichen wie Design Pattern und UML? Ich frag' deshalb, weil mir zu diesen Themen regelmäßig der Austausch fehlt. Treib' mich dann regelmäßig bei Stackoverflow rum, wobei die Community ja nicht so verständnisvoll auf didaktische Fragen reagiert.

UML und UML-Erweiterungen habe ich zum Ende meines Studiums in einigen Projekten viel mit gearbeitet, also auch praktisch.

Design Pattern relativ wenig, da eher in Gesprächen mit meinem Partner, der nach unserem Studium den klassischen Weg gegangen ist und im Bereich Softwareentwicklung arbeitet.

UML spielt aber z.B. bei denen überhaupt keine Rolle.

Also Design Patterns auf praktischer Ebene das was ich irgendwann mal in kleineren Projekten genutzt habe, aber spezielles ist irgendwo in den grauen Zellen verschwunden ist. Ich unterrichte es aktuell auch gar nicht. Müsste mich dann glaub erstmal in einige wieder rein denken. Geht mir manchmal schon bei der IHK-Korrektur so, dass ich erstmal wieder überlegen muss, wie war das nochmal...

Eigentlich schade, aber naja, der Tag hat nur 24 Stunden und wenn man sich dann auch noch mit anderen Dingen außer Programmierung oder Informatik überhaupt beschäftigen möchte.