

# Kaip informatikos mokymas gali prisidėti prie ugdymo gerinimo

Juraj Hromkovič ir Regula Lacher

Šveicarijos federalinio technologijos instituto Ciuriche (ETH Zurich) Informatikos katedra

[juraj.hromkovic@inf.ethz.ch](mailto:juraj.hromkovic@inf.ethz.ch), [regula.lacher@inf.ethz.ch](mailto:regula.lacher@inf.ethz.ch)

## Santrauka

Dabartinė švietimo sistema daugiausia dėmesio skiria faktų, modelių, metodų ir įgūdžių mokymui bei jų taikymui sprendžiant įvairius uždavinius. Kompiuterijos dėka daugumos šių uždavinių sprendimus galima automatizuoti. Ateities švietimas turi sutelkti dėmesį į pagalbą mokiniams, kad jie galėtų tyrinėti savo kūrybiškumą ir ugdyti kritinį mąstymą. Pirmiausia paaiškinsime, kaip galima paremti šios krypties plėtrą, o po to keletu pavyzdžių parodysime, kad gerai parengti informatikos vadovėliai gali atlikti novatorišką vaidmenį.

## 1. XXI amžiaus mokymo koncepcija

Pagrindinis švietimo tikslas – visapusiškas mokinių asmenybės ugdymas (atsižvelgiant į jų individualius gabumus) siekiant kuo labiau remti jų potencialo vystymąsi, taip galiausiai prisidedant prie visuomenės vystymosi. Mokytis – tai kurti naujas abstrakcijos ir modeliavimo sąvokas, kritiškai mąstyti ir būti kūrybingiems. Reikia tikėtis, kad mokyklų absolventai bus pasirengę būsimai karjerai, o platesne prasme – ateities pasauliui.

Dabar niekas negali patikimai numatyti, kokios kompetencijos bus paklausios po dvidešimties metų. Žinome tik tai, kad viskas, kas suprantama, tam tikru laipsniu bus automatizuota. Mokymasis valdyti įrangą, taikyti iš anksto parengtus metodus arba mokytis tam tikrų įgūdžių ateityje turės vis mažesnę ir mažesnę vertę. Skaitmeninės technologijos jau dabar leidžia atlikti šiuos veiksmus tiksliau, greičiau ir patikimiau. Vienintelis žmonių pranašumas palyginti su žmogaus sukurtomis technologijomis, yra kritinis mąstymas, kūrybiškumas ir su tuo susijęs gebėjimas improvizuoti, tobulinti. Todėl XXI a. mokyklos turi daugiausia dėmesio skirti mokinių emociniam ir intelektiniam vystymuisi per veiklą, skatinančią jų kūrybiškumą ir kritinį mąstymą.

Beveik visos švietimo sistemos tebėra giliai įstrigusios XIX a. pramonės revoliucijos laikų švietimo reikalavimų pinklėse. Tuo metu darbo rinkai reikėjo specialistų, gebančių taikyti sudėtingas procedūras (metodus, algoritmus), skirtus įvairioms užduotims atlikti. Tačiau dabar šie darbai iš esmės automatizuoti. Visa tai atitinka turinį šiandieninių vadovėlių, parašytų siekiant suteikti žinių ir pateikti tam tikrus įgūdžius (vadovaujantis instrukcijomis bei algoritmais). Tai tik aukštesnė įsiminimo forma. Mokykloje nebepakanka mokytis gerai žinomų sąvokų. Mokymas turi skatinti ir įgalinti mokinius vaikščioti atradimų ir naujovių keliais, iš naujo atrasti pagrindines sąvokas, metodus ir technologijas.

Dabartinės švietimo sistemos būklės neįmanoma pakeisti per trumpą laiką. Būtina pradėti evoliucinį procesą, kuris per ilgesnį laiką, žingsnis po žingsnio „švelniai“ keistų dabartinį švietimo modelį, orientuotą į iš anksto paruoštų žinių ir tam tikrų įgūdžių mokymą, į švietimą, ugdantį mokinių potencialą tose srityse, kurių nepakeis automatizavimas ir technologijos. **Mokymo ir vadovėlių kūrimo šūkis turi būti: nemokykite mokslo produktų (faktų, teiginių, metodų, modelių, technologijų) ir jų naudojimo subtilybių; aiškinkite jų atradimo procesą ir kūrybinį tobulinimą.**

Toks požiūris atitinka dabartinę tarptautinę kompetencijos įgijimo tendenciją. Kompetencija – tai ne gebėjimas spręsti įprastus uždavinius pagal išmokus metodus. Kompetencija – tai patirtis, kuria remdamiesi galime prasmingai ir originaliai veikti naujose situacijose, remdamiesi savo žiniomis ir patirtimi. Taip

mokomi absolventai bus išties novatoriški, turintys vidinę motyvaciją atrasti, suprasti ir kurti. Jie **aklai** netikės jiems pateiktais teiginiais, bet juos savarankiškai kvestionuos nagrinėdami jų kilmę ir esmę.

## Vadovėlių kūrimo ir vertinimo kriterijai

Klasikiniai bendrieji techninio teisingumo ir suprantamumo kriterijai išlieka pagrindinėmis gero mokymo sąlygomis. Tačiau šių kriterijų nepakanka minėtiems tikslams pasiekti. Raktas pradėti inovacijų procesą mokyme yra suformuluoti kriterijus, kurie padėtų autorių grupėms kurti vadovėlius, atsižvelgiant į minėtus tikslus. Visų pirma, bet kokio vadovėlis, taigi ir mokymas, turėtų būti vertinamas pagal tai, kiek jis gali padėti siekti šių tikslų:

1. Mokymosi turinio ir jo supratimo proceso skaidymas į mažų žingsnelių seką, kad mokiniai turėtų realią galimybę pereiti atskirus žingsnelius savarankiškai ir taip iš naujo atrasti ir įsisavinti žinias.

2. Išsamūs ir lengvai suprantami paaiškinimai, leidžiantys atlikti tyrimą visiškai savarankiškai arba su minimalia pagalba. Kiekvienas mokinys, atsižvelgdamas į savo poreikius, gali nagrinėti temą bet kiek kartų, bet kuriuo metu, individualiu tempu. Visų užduočių pavyzdžiai ir sprendimai apima ne tik rezultatus ir trumpą metodo aprašymą, bet ir paaiškinimus, kodėl sprendėjas elgėsi taip, o ne kitaip. Jei yra keli galimi uždavinio sprendimo būdai, jie turi būti pateikti.

3. Motyvuojantis pažintinis įvadas į kiekvieną temą, kad klasė suaktyvėtų.

4. Individuali pagalba kiekvienam mokiniui. Lengvai suprantamos pagrindinės žinios ir veikla, kurią turi įvaldyti visi, sudėtingos temos gabiems ir motyvuotiems mokiniams. Vadovėlis turi leisti mokytojui diferencijuoti tikslus pagal kiekvieno mokinio gebėjimus.

5. Mokymasis per kūrybinę veiklą. Tikslas – išmokyti analizuoti savybes, funkcionalumą ir pritaikomumą savo darbo produktų ir, atsižvelgiant į tai, motyvuoti mokinius, kad jie tobulintų jų pačių sukurtus produktus.

6. Vadovavimas mokiniams atliekant eksperimentus ir analizuojant jų rezultatus taip, kad mokiniai įgytų naujų žinių.

7. Motyvavimas ir stiprinimas noro bandyti spręsti uždavinius ir mokytis iš nesėkmingų bandymų (klaidų), nepadėjusių pasiekti užsibrėžto (-ų) tikslo (-ų).

8. Rūpestingas tikslios terminijos puoselėjimas ir tikslų formuluočių praktikavimas visose bendravimo formose.

9. Įgytų žinių tvarumo didinimas įtraukiant jas į jau turimus kontekstus.

10. Komandinio darbo skatinimas spręsti užduotis intensyviai bendraujant.

11. Siūlymas priemonių, leidžiančių mokiniams savarankiškai įvertinti savo pažangą.

Visa tai drauge turėtų garantuoti sėkmės jausmą, kad tai veiksmingiausias mokymo metodas. Galutinis tikslas – padidinti mokinių pasitikėjimą savimi ir jų norą spręsti uždavinius bei savarankiškai kurti naujus produktus.

Norint parašyti vadovėlius, kurie padėtų įgyvendinti minėtus tikslus ir priemones jiems pasiekti, reikia didelės rašančiųjų kompetencijos keliose srityse. Konkrečiau:

- išsamios dalyko žinios platesniame ir gilesniame kontekste, įskaitant nagrinėjamos mokslinės disciplinos raidą;
- pedagoginė patirtis, įgyta mokant konkretaus amžiaus grupės vaikus;
- nagrinėjamos mokslo disciplinos dalyko didaktikos žinios.

Kuriant vadovėlius rekomenduotini šie elementai:

- pažintinio aktyvinimo elementai, skirti įvadui į temą, pavyzdžiui, galvosūkiams, netikėti pastebėjimai, iš pažiūros prieštaringi teiginiai, patrauklios kompiuterių programos;
- pavyzdžiai pristatant metodus ir jų apmąstymai;

- mokymosi užduotys, kuriomis siekiama atrasti naujus ryšius (kontekstą) arba rasti uždavinio sprendimo būdą;
- nesėkmių analizė ir mokymasis iš jų;
- eksperimentų planavimas, jų vykdymas ir rezultatų analizė;
- užduočių su pageidaujamosiomis savybėmis, skirtų komandiniam darbui, rengimas;
- klasikinės pratybos, skirtos įtvirtinti tam, ko išmokta;
- istorinės ir socialinės pastabos, paaiškinančios nagrinėjamos temos atsiradimą moksliniame ir socialiniame kontekste;
- klausimai, kuriais siekiama patikrinti, ar teisingai vartojama techninė terminija;
- dažni apibendrinimai naujai įgytų žinių, atsižvelgiant į jau įgytas žinias;
- testai savarankiškai įgytoms dalyko žinioms patikrinti.

Prie kiekvieno vadovėlio turėtų būti pridedamas mokytojo vadovas, kuris taip pat turėtų būti prieinamas mokinių tėvams. Šis vadovas turėtų užtikrinti šiuos dalykus:

- teikti mokytojams žinių, susijusių su dėstomu dalyku (platesniame ir gilesniame kontekste);
- išsamiai parodyti mokytojams, kaip galima įgyvendinti pamokas, paaiškinti, tiksliai nustatyti kiekvieno vadovėlio elemento tikslą ir kaip patikrinti atitinkamo mokymosi proceso sėkmę;
- paaiškinti, kaip mokinys gali individualiai vadovautis vadovėliu mokymosi procese;
- aiškiai apibrėžti tikslus ir priemones, skirtas mokymosi pažangai įvertinti;
- pateikti išsamius pratybų sprendimus, įskaitant didaktines pastabas;
- teikti patarimus ir rekomendacijas, kaip reaguoti į mokinių siūlomus netikėtus originalius dalykus.

## 2. Informatikos mokymas kaip kitų dalykų mokymo modelis

Kodėl informatikos mokymas gali atlikti lemiamą vaidmenį keičiant švietimą pirmoje dalyje pasiūlyta kryptimi? Panagrinėkime Seymouro Paperto [20-22] konstrukcionizmą, kurio pagrindinę mintį galima išreikšti genialiu sakiniu „Mokymasis mokantis, kad viskas veiktų“. Pradedant nuo „Mokymasis kuriant žinių struktūras“ (Jean Piaget [23]), o toliau – „Mokymasis veikiant“ (Aristotelis [1] ir Komenskis [5]), mokiniai konstruoja įvairius produktus (programas, slaptus kodus, skaičių atvaizdavimus, duomenų struktūras, šifravimo funkcijas, algoritmus ir t. t.). Tačiau gaminių kūrybos užbaigimas nėra darbo su jais pabaiga, bet veikia mokymosi su jais pradžia. Čia pat naujas žingsnis – pagamintų produktų funkcionalumo ir savybių tyrimas. Ne vien tam, kad prirėkus juos pataisytų, bet ir tam, kad gautų naujų idėjų apie tai, kaip juos tobulinti arba išplėsti jų funkcionalumą. Taip mokiniai užvedami ant naujo kūrybinės veiklos kelio, įgalinančio sukurti kažką geriau nei buvo anksčiau. Būtent taip žmonės gyvena, dirba ir mokosi nuo žmonijos aušros: begalinė tobulinimo ir motyvacijos grandis, kiekvienas žingsnis gilina supratimą ir didina kompetenciją. Tai vienintelis būdas tinkamai ugdyti kompetencijas.

Toliau aptarsime tik dvi su programavimo mokymu susijusias sąvokas ir duomenų saugumą. Daugybė tokių mokymo pavyzdžių yra vadovėliuose mokiniams [2, 3, 8, 9, 12, 13, 15, 19] ir mokytojams [4, 6, 7, 10, 11, 14, 16, 18], apimančių visas amžiaus grupes nuo vaikų darželio iki gimnazijos.

### Programavimas

Kodėl programuojame? Todėl, kad norime apibūdinti tam tikrą veiklą kompiuteriui arba robotui taip, kad šie techniniai įrenginiai galėtų vykdyti šią veiklą savarankiškai. Kompiuterio programa yra veiklos aprašymas programavimo kalba, suprantama kompiuteriui. Programavimas – labai kūrybinga ir konstruktyvi veikla, atitinkanti pagrindinius mūsų tikslus, nes:

- Reikia rasti strategiją, kaip pasiekti tam tikrą tikslą. Taigi, uždavinio sprendimas yra pagrindinis klausimas. Pirmiausia mokinys gali mokytis teisingai suprasti uždavinio aprašymą, tada sukurti savą uždavinio aprašymą naudodamas įvairias priemones – lenteles, grafikus, lygtis ir t. t. – ir taip

sukaupiti patirtį bandydamas spręsti konkrečius uždavinius. Sukūręs sprendimo strategiją, mokinys gali išbandyti jos funkcionalumą ieškodamas atvejų, kuriems jo strategija neveikia, arba ieškoti argumentų, kaip ir kodėl tinkamai veikia kitais atvejais. Tai modelio tikrinimas eksperimentuojant.

- Mokiniai, sukūrę algoritmą (sprendimo strategiją), turi aprašyti algoritmą programavimo kalba. Programa gali būti analizuojama kaip pačių mokinių darbo produktas sutelkiant dėmesį į jos funkcionalumą. Tai viena iš priežasčių, kodėl programavimo naujokams tinka užduotys, susiję su judėjimu plokštumoje arba geometrinių figūrų piešimu, nes tada galima vizualiai ištirti programos vykdymą. Jei programa neveikia tinkamai, mokiniai turi bandyti ištaisyti programoje esančias logines klaidas, peržiūrėti savo strategiją. Jei programa veikia tinkamai, mokiniai gali galvoti apie jos funkcionalumo išplėtimą arba apie sudėtingesnius uždavinius ir naudoti sukurtą programą kaip modulį (statybinį bloką) naujai programai, skirtai sudėtingesniems uždaviniams, kurti.
- Pažengę mokiniai gali analizuoti savo programas, atsižvelgdami į jų veikimo sudėtingumą, palyginti skirtingų programų veiksmingumą ir tada pradėti ieškoti algoritmo arba jo įgyvendinimo, kuris būtų efektyvesnis už iki šiol sukurtus.
- Norint pasiekti aukštą programuotojo kompetenciją, nepakanka kurti programas skirtingiems uždaviniams spręsti. Mokiniai turi būti mokomi analizuoti kitų sukurtas programas, jas taisyti ir modifikuoti (pritaikyti) naujiems uždaviniams. Visa tai vėlgi reiškia daug bandymų (eksperimentinio darbo), kad būtų suprastas tiriamos programos funkcionalumas.
- Programa – tai sintaksiškai taisyklingas tekstas, užrašytas programavimo kalba. Kompiuteris gali vykdyti programą tik tada, kai joje nėra sintaksinių (gramatinių) klaidų. Programa yra tikslus ir išsamus veiklos aprašymas, o programos tekstas yra vienareikšmis (turintis unikalią interpretaciją). Taigi, programų rašymas yra geras mokymas taikant sintaksės taisykles ir semantiškai teisingai išreiškiant, ką norėtusi pasakyti.

Pastebėsime, kad 1 skyriaus 1, 2, 5, 6, 7 ir 10 skyreliuose išvardyti gero mokymo tikslai yra netiesiogiai įtraukti į gerai parengtus programavimo kursus. Be to, visus likusius tikslus galima pasiekti tada, kai mokymo eiga tinkamai suprojektuota. Mes atsižvelgėme į pirmąjį tikslą naudodami istorinį metodą, t. y., programavimo kalbų atsiradimą ir plėtotę (genezę). Mokiniai pradeda mokytis programuoti naudodami labai mažą žodyną. Mes mokome juos, kaip kurti naujus žodžius ir paaiškinti jų reikšmę kompiuteriui. Tokiu būdu mokiniai dalyvauja kuriant programavimo kalbą, tobulina jos raišką ir ją išbando programuodami savo uždavinius.

## Kriptografija

Kriptografijai – slaptų raštų rengimui ir analizei – yra apie 4000 metų. Tai duomenų saugumo pagrindas. Vadovaudamiesi pirmuoju gero mokymo tikslu vėl siūlome taikyti istorinį metodą. Pagrindinis tikslas yra ne parodyti slaptų užrašų ar kriptosistemų pavyzdžių ir išmokyti jais naudotis, bet pateikti mokiniams keletą metodų, kaip kurti ir iššifruoti slaptus raštus, skatinti kurti savo originalius slaptus raštus ir bandyti „nulaužti“ kitų sukurtas sistemas.

Šią temą galima pradėti mokytis labai anksti (jau nuo trečios klasės), nes žinoma, kad senovėje buvo taisyklė, jog slaptų raštų aprašymas turi būti toks paprastas, kad kiekvienas galėtų jį išmokti mintinai, t. y., aprašymo nereikėtų išsaugoti rašytine forma. Iš [3, 6, 7, 11, 12, 15, 16] matyti, kad pirmieji slapti raštai pagrįsti transpozicijomis, jas galima pavaizduoti paprastais paveikslėliais, todėl tokią sistemą (šifravimo raktą) mokinys gali susikurti pats. Tas pats pasakytina ir apie pakeitimus pagrįstą slaptą rašymą. Šiuo atveju mokiniai gali pafantazuoti kurdami naujas abėcėles.

Sukurtų kriptosistemų savybės (kokybė) tiriamos jas atpažįstant. Pirmiausia mokiniai gali išmokti nulaužti Cezario kriptosistemą analizuodami raidžių dažnį šifruotuose tekstuose, o vėliau – bet kokią monoabėcėlinę kriptosistemą analizuodami raidžių dažnius. Mėgindami užtikrinti savo kriptosistemų apsaugą nuo raidžių dažnio analizės, mokiniai gali iš naujo sukurti Viženero (Blaise de Vigenère) kriptosistemą.

Norint šia kryptimi eiti mažais žingsniais, vėlgi galima sekti istorinę raidą, kurioje pirmas žingsnis buvo vienetu padidinti Cezario raktą (poslinkį abėcėlėje), iššifravus kiekvieną raidę. Viženėro šifravimas yra ne kas kita, kaip šokinėjimas „tabula recta“ lentele pagal specialų modelį. Kitas paprastas žingsnis link Viženėro šifro yra naudoti skirtingus Cezario raktus lyginėms ir nelyginėms raidėms grynajame tekste. Galima vėlgi leisti mokiniams patiems kurti daugiaabėcėlines kriptosistemas prieš artėjant prie Viženėro šifravimo būdo. Įveikti Viženėro šifrą taip pat galima mažų žingsnelių seka [3] ir gali būti kvietimas pereiti nuo kontekstinių kriptosistemų prie kontekstinių (blokais pagrįstų).

Čia yra daug laisvės kurti daugybę savų kriptosistemų, kurios negali būti nulaužtos naudojantis Viženėro šifro analize. Čia ir baigiame, nes aprašyti, kaip šiuolaikinę kriptografiją padaryti suprantamą vidurinės mokyklos mokiniams reikėtų žymiai daugiau vietos, negu čia turime. Trumpai pažinčiai su viešojo rakto kriptografija vidurinėse mokyklose rekomenduojame straipsnį [17]. Bet svarbiausia yra tai, kad galime mokytis slaptaraščio taip, kad didžiąją dalį esminių sąvokų mokiniai galėtų įsisavinti patys, o kitas mažomis porcijomis. Kita priežastis, kodėl kriptografija labai patraukli pamokoms yra ta, kad kiekvienos naujos temos pradžioje klasės dėmesį galima patraukti linksmomis ir intriguojančiais galvosūkiomis.

### 3. Išvada

Pateikėme dvi pagrindines informatikos temas (programavimą ir kriptografiją) tam, kad parodytume, jog jos yra svarbios tuo, kad gali būti panaudotos kaip modeliai (pavyzdžiai) edukacijos tikslams pasiekti kitose disciplinose. Svarbu, kad tokiu būdu gali būti mokoma ir kitų informatikos temų.

Pagrindinė informatikos mokslininkų veikla:

- objektų atvaizdų ir kodų su reikiamomis savybėmis (trumpas, slaptas, save tikrinantis, efektyvus apdorojimas ir t. t.) **kūrimas**;
- efektyvių algoritmų įvairiems tikslams (uždavinių sprendimo, kodų ir kalbų vertimo ir t. t.) **projektavimas**;
- programavimo technologijų ir informacinių technologijų, teikiančių daugybę paslaugų, **valdymas**.

Visais šiais atvejais ekspertizę galime atlikti taip, kaip čia siūloma. Kas nors suprojektuoja ir sukuria gaminį, kurio savybes galima analizuoti funkcionalumo ir tinkamumo skirtingoms reikmėms požiūriu. Gali būti taikomas istorinis metodas siekiant, kad visos žinios būtų pasiekiamos mažais žingsniais, pagrįstais besimokančiųjų veikla.

### Padėkos

Už įdomias diskusijas ir pastabas ankstesnėms šio straipsnio versijoms dėkojame šiems asmenims: Peter Borovansky, Martin Curuc, Dennis Komm, Viliam Kratochvil, Daniel Masarovič, Maria Smerkova, Gabriel Thullen, Michal Winczer.

### Literatūra

1. Aristotle. A New Translation of the Nichomachean Ethics of Aristotle. J. Vincent. Oxford, 1835
2. Jarka Arnold, Cedric Donner, Urs Hauser, Matthias Hauswirth, Juraj Hromkovič, Tobias Kohn, David Maletinsky, and Nicole Roth. Informatik, Programmieren und Robotik (in German). Klett und Balmer. Baar, 2021.
3. Michael Barot, Britta Dorn, Ghislain Fourny, Jens Gallenbacher, Juraj Hromkovič, and Regula Lacher. Informatik, Data Science und Sicherheit (in German). Klett und Balmer. Baar, 2021.
4. Michelle Barnett, Juraj Hromkovič, Anna Laura John, Regula Lacher, Pascal Lutscher, and Jacqueline Staub. Einfach Informatik 1 / 2, Spielerisch Programmieren mit Robotern (in German). Klett und Balmer. Baar, 2021.
5. John A. Comenius. Didactica magna. 1675.

6. Urs Hauser, Juraj Hromkovič, Petra Klingenstein, Regula Lacher, Pascal Lutscher, and Jacqueline Staub. Einfach Informatik 1 / 2, Ratsel und Spiele ohne Computer (in German). Klett und Balmer. Baar, 2021.
7. Heinz Hofer, Juraj Hromkovič, Regula Lacher, Pascal Lutscher, and Urs Wildeisen. Einfach Informatik 3 / 4. Programmieren und Ratsel lösen, teachers guide (in German). Klett und Balmer. Baar, 2021.
8. Heinz Hofer, Juraj Hromkovič, Regula Lacher, Pascal Lutscher, and Urs Wildeisen. Einfach Informatik 3 / 4. Programmieren und Ratsel lösen, textbook (in German). Klett und Balmer. Baar, 2021.
9. Juraj Hromkovič. Einfach Informatik 5 / 6. Programmieren, textbook (in German). Klett und Balmer. Baar, 2018.
10. Juraj Hromkovič. Einfach Informatik 5 / 6. Programmieren, teacher's guide (in German). Klett und Balmer. Baar, 2019.
11. Juraj Hromkovič and Regula Lacher. Einfach Informatik 5 / 6. Losungen finden, teacher's guide (in German). Klett und Balmer. Baar, 2019.
12. Juraj Hromkovič and Regula Lacher. Einfach Informatik 5 / 6. Losungen finden, textbook (in German). Klett und Balmer. Baar, 2019.
13. Juraj Hromkovič. Einfach Informatik 7–9. Strategien entwickeln, textbook (in German). Klett und Balmer. Baar, 2018.
14. Juraj Hromkovič. Einfach Informatik 7–9. Strategien entwickeln, teacher's guide (in German). Klett und Balmer. Baar, 2018.
15. Juraj Hromkovič. Einfach Informatik 7–9. Daten darstellen, verschlüsseln, komprimieren, textbook (in German). Klett und Balmer. Baar, 2018.
16. Juraj Hromkovič. Einfach Informatik 7–9. Daten darstellen, verschlüsseln, komprimieren, teacher's guide (in German). Klett und Balmer. Baar, 2018.
17. Lucia Keller, Dennis Komm, Giovanni Serafini, Andreas Sprock, and Bjorn Steffen. Teaching public-key cryptography in school In Proc. of the 4th International Conference on Informatics in Secondary Schools: Evolution and Perspectives (ISSEP 2010), volume 5941 of Lecture Notes in Computer Science, Springer-Verlag 2010, pages 112–123.
18. Tobias Kohn and Juraj Hromkovič. Einfach Informatik 7–9. Programmieren, teacher's guide (in German). Klett und Balmer. Baar, 2018.
19. Tobias Kohn and Juraj Hromkovič. Einfach Informatik 7–9. Programmieren, textbook (in German). Klett und Balmer. Baar, 2018.
20. Seymour Papert. Mindstorms: Children, Computers and Powerful Ideas. Basic Books, Inc. Publ. New York, 1980.
21. Seymour Papert. The Children's Machine: Rethinking School in the Age of the Computer. Basic Books, Inc. Publ. New York, 1993.
22. Seymour Papert and David Harel. Constructionism. Ablex publishing. New York, 1991.
23. Jean Piaget and David Harel. The Psychology of Intelligence. Cambridge, Harvard University Press, 1950.

Šį straipsnį [J. Hromkovic., R. Lacher. "How Teaching Informatics Can Contribute to Improving Education in General." *Bulletin of the European Association for Theoretical Computer Science*, 2023, 139(1). <http://bulletin.eatcs.org/index.php/beatcs/article/view/752>] iš anglų kalbos vertė Gintautas Grigas.