

# Opinia o nowej podstawie programowej zajęć komputerowych w szkole podstawowej oraz informatyki w gimnazjum i szkołach ponadgimnazjalnych

19

Dr Andrzej Walat

## 1. Uwagi ogólne

Podstawa programowa zajęć komputerowych i informatyki obejmuje faktycznie dwa zakresy kształcenia *informatykę* oraz *technologię informacyjną (i komunikacyjną)*. Jeszcze do niedawna, w dyskusjach oświatowych, bardzo często mylono te dwa zakresy i dlatego uważam za osiągnięcie, że już się ich nie myli. Widać to zarówno w nowej podstawie, jak i szczególnie w komentarzu (do nowej podstawy programowej) profesora Sysły. Jednak nadal dyskusyjne są kryteria rozróżnienia, ściśle definicje tych dwóch dyscyplin i zakresów kształcenia oraz do pewnego stopnia ich polskie nazwy. Rozróżnienie w komentarzu profesora Sysły wydaje mi się może trochę uproszczone, chociaż bardzo doceniam zwrócenie uwagi na fakt, że „w informatyce liczy się produkt”, bo to bardzo istotna cecha informatyki i będę do tego jeszcze wracał w tej opinii.

Bardziej odpowiadają mi definicje sformułowane krótko i dobitnie przez zespół ekspertów ACM w raporcie [1], które lepiej charakteryzują obie dyscypliny i różnice między nimi. Zwracają oni uwagę na dwie podstawowe cechy różnicujące. Po pierwsze, informatyka jest nauką podstawową, a technologia informacyjna stosowaną (co oczywiście nie wyklucza tego, że informatyka ma liczne i ważne zastosowania praktyczne). Po drugie, te dwie dyscypliny mają różny przedmiot zainteresowań. Informatyka zajmuje się złożonymi procesami algorytmicznymi i komputerowymi. Technologia informacyjna tym, w jaki sposób ludzie skutecznie posługują się i dzielą informacją reprezentowaną w różnej formie: tekstowej, dźwiękowej, wizualnej, video, itd. To rozróżnienie wydaje mi się kluczowe.

Jeśli chodzi o nazwy, to decyzja, by dwuwyrazową nazwę *technologia informacyjna* zastąpić bardziej

precyzyjną i odpowiadającą aktualnej rzeczywistości dłuższą nazwą *technologia informacyjna i komunikacyjna*, wydaje mi się zasadniczo słuszną, ale może trochę niepraktyczną. Konsekwencją przyjęcia takiej nazwy jest zwykle to, że w wielu publikacjach, w tym także w komentarzu profesora Sysły, po zasadniczo ideowo słusznym uzasadnieniu pełniejszej nazwy, następuje zapowiedź: „dalej w tym komentarzu używamy czasem krótkiego określenia technologia”.

Tę walkę racji z wygodą (IT – *information technology* z ICT – *information and communication technology*) obserwuję w literaturze światowej już od wielu lat i nie przewiduję, by się szybko i ostatecznie rozstrzygnęła. Zwykle każde zwycięstwo jest lokalne i krótkoterminowe.

Jeśli chodzi o informatykę, to bardzo ważne wydaje mi się zwrócenie przez profesora Sysłę uwagi na fakt, że wraz z rozwojem informatyki zwiększają się oczekiwania wobec edukacji. Do niedawna przyjmowaliśmy, że jednym z najważniejszych celów edukacji informatycznej jest rozwijanie *myślenia algorytmicznego*, dzisiaj to już nie wystarcza i chodzi o coś więcej – rozwijanie *myślenia komputacyjnego* (ang. *computational thinking*). Do tej ważnej uwagi będę jeszcze wracał. W tym miejscu nie mogę nie przypomnieć, że wiele lat temu zaproponowałem, bez wielkich szans na powodzenie, by polską nazwę dyscypliny i przedmiotu nauczania *informatyka* zmienić na bardziej moim zdaniem odpowiednią – *komputyka*. Gdyby taka propozycja została przyjęta, to zwolnione słowo (tylko jedno) – *informatyka* mogłoby zastąpić wielowyrazowe nazwy *technologia informacyjna (i komunikacyjna)*. Warto zwrócić uwagę na fakt, że definicja technologii informacyjnej w raporcie ACM jest bardzo podobna (wręcz taka sama) do definicji informatyki w bardzo wielu tradycyjnych podręcznikach.

Oceniam jako postęp odejście – mam nadzieję, że ostateczne – od obowiązujących jeszcze niedawno błędnych koncepcji, że informatyka w szkołach ponadgimnazjalnych jest programowym rozszerzeniem technologii informacyjnej, oraz że wiedza i umiejętności informatyczne są potrzebne tylko nielicznym, a większości wykształconych ludzi wystarczy znajomość technologii informacyjnej.

Dobrze, że w szkołach ponadgimnazjalnych informatyka, przynajmniej na poziomie podstawowym, jest ostatecznie przedmiotem dla wszystkich, oraz że zajęcia komputerowe zaczynają się w szkole podstawowej już od pierwszej klasy.

Generalnie oceniam nową podstawę programową edukacji informatycznej raczej dobrze, zwłaszcza w porównaniu z dotychczas obowiązującą i poprzednią, ale nie bez (nawet dość poważnych) zastrzeżeń. Odnoszą się one w różnym stopniu do różnych etapów nauczania i charakteryzują je etapami.

## 2. Zajęcia komputerowe w szkole podstawowej

Nowa podstawa programowa zajęć komputerowych w szkole podstawowej wydaje mi się bardzo dobra. Nie mam do niej żadnych zastrzeżeń. Podzielim entuzjazm profesora Sysły dla decyzji oficjalnego uznania potrzeby zajęć komputerowych już na pierwszym etapie nauczania (w klasach 1-3). Bardzo podoba mi się zapis (zawarty w podstawie zajęć komputerowych dla drugiego etapu nauczania): „uczeń ... za pomocą ciągu poleceń tworzy proste motywy lub steruje obiektem na ekranie”.

Ten zapis według komentarza prof. M. Sysły „dotyczy wykorzystania Logo”, ale jego główną zaletą jest to, że daje podstawę do propedeutycznego kształcenia myślenia algorytmicznego na odpowiednio wczesnym etapie nauczania. Jestem przekonany, że dzięki rozsądnie prowadzonym zajęciom w szkole podstawowej z zakresu *tworzenia prostych motywów za pomocą ciągu poleceń* (co, jak pokazują doświadczenia szkolne bardzo wielu nauczycieli, jest atrakcyjne i kształcące), elementarne programowanie na następnym etapie – gimnazjum, w odpowiednio przyjaznym i dostosowanym do wieku środowisku programowania, nie będzie za trudne.

## 3. Informatyka w gimnazjum

Do nowej podstawy programowej informatyki w gimnazjum mam już kilka zastrzeżeń, a oto moje uwagi.

### 3.1. Czy nie za mało informatyki w podstawie programowej przedmiotu informatyka?

Całkowicie zgadzam się z następującą opinią i sugestią profesora Sysły:

„Wydaje się, że kształcenie w zakresie rozszerzonym zaczyna się zbyt późno w szkole ponadgimnazjalnej. Dlatego sugeruje się, by dobrze wykorzystać zajęcia na informatyce w gimnazjum i w liceum (w zakresie podstawowym).”

Ale jeśli powyższą sugestię mamy potraktować poważnie, to w podstawie programowej informatyki w gimnazjum i szkołach ponadgimnazjalnych (na poziomie podstawowym) powinno być choć trochę więcej informatyki. Aktualnie jest jej znikomą małą – jest prawie wyłącznie technologia informacyjna.

W szczególności, powinny wystąpić w rozsądnym zakresie elementy programowania w odpowiednim, dostosowanym do wieku języku i środowisku. Zapisy w ustępie 5 wymagań szczegółowych *Rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera, stosowanie podejścia algorytmicznego*, jedynym działem, w którym byłoby miejsce na elementy programowania, uważam za niewystarczające. Moim zdaniem podpunkt 2, który obecnie ma treść:

*(uczeń) formułuje ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmicznej;*

lepiej brzmiałby w formie rozszerzonej:

*(uczeń) formułuje ścisły opis prostej sytuacji problemowej, analizuje ją i przedstawia rozwiązanie w postaci algorytmu, zapisuje algorytm w postaci programu (procedury) wykonywalnego przez komputer, ocenia i weryfikuje praktycznie swoje rozwiązanie uruchamiając program dla wybranych danych testowych.*

W związku z tą zmianą można byłoby usunąć podpunkt 5 o treści:

*(uczeń) ... wykonuje wybrane algorytmy za pomocą komputera* (bo przecież algorytmy ma wykonywać komputer, a uczeń je zapisać w postaci w odpowiednim języku i uruchomić).

To rozszerzenie, jak dowodzą tego bardzo bogate doświadczenia nauczycieli pracujących z młodzieżą gimnazjalną, a nawet szkół podstawowych,

w żadnym wypadku nie będzie wymaganiem wygórowanym. Natomiast ma ono bardzo ważne uzasadnienia.

Po pierwsze, chociaż informatyka to nie tylko programowanie, jest ono niewątpliwie podstawową aktywnością informatyczną. Trudno poczuć smak informatyki bez choćby liźnięcia programowania.

Jak słusznie podkreśla profesor Sysło, istotną cechą informatyki jest to, że informatycy tworzą produkty, a te produkty to głównie programy. Warto uczyć od małego, że w informatyce nie wystarczy opowiedzieć „jak to można zrobić”, ale trzeba zrobić i w praktyce sprawdzić. Praktyczna implementacja rozwiązania algorytmicznego jest równie ważna, jak wymyślenie/dobranie odpowiedniego algorytmu.

Po drugie, obawiam się, że pozbawienie uczniów możliwości praktycznej realizacji i weryfikacji ich idei algorytmicznych, bardzo osłabi atrakcyjność zajęć z informatyki. Każdy nauczyciel, który widział uczniów wyskakujących z uniesionymi rękami do góry i okrzykiem JEEEST!, w reakcji na pomyślnie uruchomienie programu, wie dobrze o co mi chodzi. Nie chciałbym pozbawiać uczniów takich przeżyć.

Po trzecie, programowanie jest nie tylko zajęciem dla informatyków, ale może być także ważną aktywnością poznawczą, istnieje ogromna literatura pokazująca, jak bardzo skuteczne może być uczenie się przez programowanie.

### 3.2. Wybór odpowiednich przykładów klasycznych algorytmów, jakie powinien poznać uczeń gimnazjum

Uczeń na lekcjach informatyki powinien niewątpliwie poznać wybrane klasyczne algorytmy i jest to zapisane w podstawie programowej. Bardzo dobrze, że lista klasycznych algorytmów wymienionych w wymaganiach szczegółowych jest krótka, a przykłady dobrane są trafnie. Są to rzeczywiście problemy i algorytmy najbardziej reprezentatywne dla dyscypliny, mające duże znaczenie praktyczne, a jednocześnie zrozumiałe dla gimnazjalisty.

Jednak, choć bardzo doceniam wykazany w tym wypadku umiar, do listy klasycznych algorytmów, które powinien poznać uczeń gimnazjum dopisałbym jeszcze: algorytm Euklidesa, algorytm Herona obliczania pierwiastka kwadratowego oraz algorytmy zamiany dziesiętnych zapisów liczb na dwójkowe i odwrotnie, a to z kilku bardzo istotnych powodów.

Po pierwsze, te przykłady pokazują, że ważne i mające do dziś znaczenie algorytmy, tworzono już w starożytności. To pozwala lepiej poznać i zrozumieć historię informatyki (komputyki) – to nie jest tylko historia komputerów i maszyn liczących.

Po drugie, wskazane przeze mnie przykłady pozwalają lepiej zrozumieć związki informatyki z innymi dyscyplinami, w tym wypadku z matematyką. Informatyka ma nie tylko (jak każda dyscyplina) swoje specyficzne problemy, ale pomaga też w rozwiązywaniu problemów i w lepszym zrozumieniu ważnych zagadnień (jak np. system pozycyjny) z innych dziedzin. To jest powód, dla którego wszyscy powinni się uczyć informatyki, a nie tylko kandydaci na przyszłych informatyków.

Po trzecie, takie przykłady inspirują do bardziej interdyscyplinarnego nauczania, zachęcają do współpracy nauczycieli różnych przedmiotów. Algorytmy Euklidesa i Herona były kiedyś (dwadzieścia lat temu) w szkolnych podręcznikach do matematyki dla klasy 5 szkoły podstawowej. Obecnie algorytm Euklidesa można spotkać w niektórych podręcznikach do 3 klasy gimnazjum. Ale zwykle kończy się tylko na opisie słownym, czasem niepoprawnym. Zapisanie algorytmu w języku „rozumiałym dla komputera” i wykonanie go dla wybranych danych testowych, stwarza zupełnie nową sytuację dydaktyczną. Jeśli uczeń nieprecyzyjnie rozumie algorytm, to źle go zapisze i praktyczne doświadczenie – uruchomienie programu – wykaże błąd. Uczeń może sam zauważyć i poprawić błąd. Uczenie się na błędach jest bardzo skuteczne.

W związku z algorytmami Euklidesa i Herona, a także algorytmami sortowania i wyszukiwania nie sposób nie wspomnieć o rekurencji. Wiele lat temu hasło *rekurencja* występowało w podstawie programowej informatyki dawnej ośmioletniej szkoły podstawowej. Teraz pojawia się po raz pierwszy w podstawie programowej informatyki dla szkół ponadgimnazjalnych, dopiero na poziomie rozszerzonym. Uważam, że to duży krok wstecz. W ciągu ostatnich lat zrozumienie znaczenia rekurencji jako podstawowej techniki rozwiązywania złożonych problemów bardzo znacznie wzrosło. Jest to hasło ponadprzedmiotowe, występujące nie tylko w programach nauczania informatyki, ale również w bardzo wielu nowoczesnych programach edukacji matematycznej i to już na niskich etapach nauczania. Uważam, że uczeń gimnazjum powinien rozumieć rekurencję i umieć z niej korzystać w rozwiązywaniu umiarkowanie złożonych problemów.

### 3.3. Niewystarczająca liczba godzin

Liczba godzin na informatykę w gimnazjum nie powinna być mniejsza niż na zajęcia komputerowe na drugim etapie nauczania, bo zakres i poziom wymagań jest znacznie większy.

### 4. Informatyka w szkołach ponadgimnazjalnych na poziomie podstawowym

Na wstępie w tej opinii wyraziłem zadowolenie, że w szkołach ponadgimnazjalnych jest informatyka na poziomie podstawowym dla wszystkich, zamiast, jak to było jeszcze niedawno, technologii informacyjnej. Wykształconemu człowiekowi niewątpliwie potrzebna jest wiedza i umiejętności z obu tych dziedzin: zarówno z technologii informacyjnej, jak i informatyki, ale opracowywania tekstów, przygotowywania prezentacji, tworzenia dzieł plastycznych, modelowania zjawisk przyrodniczych z wykorzystaniem technologii informacyjnych itd. może uczyć odpowiednio przygotowany polonista, plastyk, fizyk lub biolog na lekcjach swojego przedmiotu, natomiast podstaw informatyki musi uczyć specjalista informatyk na lekcjach informatyki.

Niestety bliższa lektura nowej podstawy wskazuje, że zmiana w porównaniu z poprzednią sytuacją polega tylko na zmianie nazwy przedmiotu. W nowej podstawie faktycznie prawie nie ma informatyki; jest to klasyczny program przedmiotu technologia informacyjna. Tej oceny nie mogą zmienić nieliczne zapisy o wykorzystaniu podejścia algorytmicznego w rozwiązywaniu problemów. Według raportu ACM [1] biegłość w posługiwaniu się technologią (co jest celem kształcenia w zakresie technologii informacyjnej) oznacza w szczególności korzystanie z myślenia algorytmicznego, w tym także umiejętność programowania.

### 5. Informatyka w szkołach ponadgimnazjalnych na poziomie rozszerzonym

W podstawie programowej nauczania informatyki w szkołach ponadgimnazjalnych jest bardzo dużo szczegółowych wymagań z różnych obszarów informatyki. Wymagana jest znajomość bardzo dużej liczby klasycznych algorytmów, umiejętność projektowania i praktycznej realizacji relacyjnych baz danych, do tego dochodzi bardzo długa lista wymagań z zakresu posługiwania się TI. Chociaż wszystkie wymienione w wymaganiach algorytmy i umiejętności są ważne, obawiam się, że w ogra-

nicznej liczbie godzin na informatykę nie da się tego w rozsądnym stopniu opanować. Nie będzie też czasu na samodzielne poszukiwania uczniów i uczenia się na przykładach różnych metod rozwiązywania problemów (nie tylko informatycznych).

Pomimo tego natłoku szczegółowych haseł, w nowej podstawie brak ważnych zapisów, które moim zdaniem, koniecznie powinny się tam znaleźć.

W komentarzu profesora Sysły do nowej podstawy programowej znajduję następujące ważne wyjaśnienie, co to jest myślenie komputacyjne.

W ostatnich latach analiza wyzwań stawianych przed obywatelami rozwijających się społeczeństw, w których gospodarka bazuje na wiedzy, doprowadziła do dalszego rozszerzenia niezbędnego zakresu kompetencji informatycznych na **myślenie komputacyjne** (ang. *computational thinking*), które obejmuje szeroki zakres intelektualnych narzędzi, reprezentujących spektrum informatycznych metod modelowania i rozwiązywania problemów, znacznie szerszy niż tradycyjnie rozumiane myślenie algorytmiczne, takich na przykład, jak:

- redukcja i dekompozycja złożonego problemu, aby móc go rozwiązać efektywnie;
- aproksymacja, gdy dokładne rozwiązanie jest poza zasięgiem nawet komputerów;
- rekurencja jako metoda indukcyjnego myślenia i zwięzłej, komputerowej implementacji rozwiązań;
- modelowanie wybranych aspektów złożonych problemów;
- znajdowanie rozwiązań metodami heurystycznymi.

Ale w nowej podstawie programowej, nie znajduję zapisów, które by takie komputacyjne myślenie (szersze i bardziej otwarte niż tradycyjnie rozumiane myślenie algorytmiczne) kształtowały.

W przeszłości, w informatyce przez wiele lat dominowały problemy obliczeniowe. Głównym zadaniem informatyków było pisanie programów, które dla określonych danych dawały odpowiednie poprawne wyniki. To te doświadczenia doprowadziły do powstania metodologii rozwiązywania problemów informatycznych, której istotnym elementem jest specyfikacja problemu składająca się z: opisu danych, opisu wyników oraz opisu relacji między danymi i wynikami.

Powstała również metodologia programowania nazywana *programowaniem imperatywnym*. Z czasem pojawiły się i zaczęły nabierać znaczenia nowe rodzaje zadań programistycznych, a w związku z tym powstały nowe podejścia do problemów informatycznych – nowe metodologie i języki programowania. Rozwój badań nad sztuczną inteligencją był

impulsem do powstania metod i języków *programowania funkcyjnego*. Wzrost znaczenia zadań sterowania, w których nie chodzi o obliczanie określonych wyników dla określonych danych, ale o gładkie sterowanie współdziałaniem wielu obiektów, doprowadził do powstania *programowania zorientowanego obiektowo*. Niestety, o tym nie dowie się uczeń na lekcjach informatyki według nowej podstawy programowej. Obraz informatyki w nowej podstawie, jest moim zdaniem, trochę zbyt statyczny, a jest to przecież dyscyplina, która się dynamicznie rozwija, nieustannie rodzą się nowe wielkie idee (paradygmaty), nie tylko nowe technologie.

W większości znanych mi współczesnych programów nauczania informatyki dla ambitnych licealistów, absolutnie dominujący jest paradygmat programowania zorientowanego obiektowo. Istnieją języki i zintegrowane środowiska programowania dostępne bezpłatnie, przyjazne i przystępne, umożliwiające gładkie wejście w zagadnienia programowania obiektowego już uzdolnionym gimnazjalistom. Istnieje również bardzo wiele materiałów dydaktycznych i publikacji pokazujących, jak można tego skutecznie uczyć na elementarnym poziomie. Wprowadzenie elementów programowania obiektowego do programu nauczania informatyki istotnie zwiększy możliwości kształtowanie myślenia komputacyjnego, elastycznego, otwartego stosunku do rozwiązywania problemów, w którym jest miejsce zarówno na tradycyjną metodologię (analiza problemu, specyfikacja, zaprojektowanie rozwiązania, itd. ), jak i na inne podejścia tworzenie hierarchii klas jako modelu wybranego fragmentu rzeczywistości, heurystykę, itd.

Oczywiście, nie można ominąć pytania: czy jest na to czas, czy jest możliwa realizacja takich ambitnych zamierzeń. Istnieje kilka możliwych rozwiązań problemu braku wystarczającej liczby godzin na naukę informatyki.

Po pierwsze, można w podstawie programowej informatyki ograniczyć liczbę wymagań dotyczących umiejętności posługiwania się technologiami informacyjnymi (i komunikacyjnymi). Informatyka i technologia informacyjna wyróżniają się spośród innych dyscyplin – przedmiotów szkolnych – tym, że uczniowie znaczną część wiedzy i umiejętności zdobywają samodzielnie w czasie pozalekcyjnym. Dotyczy to zwłaszcza uczniów szczególnie zainteresowanych informatyką, którzy wybierają informatykę na poziomie rozszerzonym. Proponuję, by okazać im więcej zaufania i założyć, że posługiwania się technologią informacyjną potrafią nauczyć się sami. Często już w momencie, gdy rozpoczynają naukę w liceum, potrafią w tym zakresie więcej niż wymaga tego podstawa programowa informatyki na poziomie rozszerzonym. Skreślenie znacznej liczby haseł z podstawy dałoby więcej czasu nauczycielom,

którzy są zobowiązani do tego, by każdy zapisany w podstawie temat skrupulatnie realizować.

To nie oznacza, że uczniowie na lekcjach informatyki, w ramach realizacji różnych projektów informatycznych, nie będą redagowali tekstów, tworzyli rysunków, itd., ale nie będzie to już tematem lekcji, lecz wykorzystywaniem (przy okazji) i doskonaleniem umiejętności zdobytych wcześniej lub samodzielnie w czasie pozalekcyjnym.

Po drugie, warto spróbować tak ukształtować podstawę programową, by nie wszyscy musieli się uczyć „równym frontem” tych samych umiejętności w tym samym zakresie, ale by każdy uczeń miał możliwość pogłębiania wiedzy w wybranym obszarze, odpowiednio do osobistych ambicji i zainteresowań. Wyobrażam sobie, że pewna część licealistów może być szczególnie zainteresowana rozwiązywaniem problemów algorytmicznych zbliżonych charakterem do zadań olimpijskich. Wystarczy im solidna klasyczna algorytmika. Inna grupa może być bardziej zainteresowana projektowaniem relacyjnych baz danych z wykorzystaniem języka SQL, a inna realizacją interaktywnych projektów dydaktycznych, gier, programów symulacyjnych, itp. z wykorzystaniem metod i środowisk programowania obiektowego i nie warto ich gnębić nadmiernymi wymaganiami w zakresie tradycyjnej algorytmiki.

Odpowiedź na pytanie, czy przyjęcie zaproponowanych wyżej rozwiązań jest możliwe i pożądane, jest silnie uzależnione od naszych wyobrażeń o tym, jaka powinna być oświata i szkoła. Czy ma być jak słynny walec, który „przyjdzie i wyrówna”, czy też ma dawać każdemu uczniowi rzeczywiste szanse rozwoju jego osobistych zainteresowań i realizacji jego aspiracji. W teorii często opowiadamy się za drugą opcją, ale w praktyce częściej wybieramy pierwszą. Osobiście jestem za konsekwentnym wyborem drugiej opcji.

### Cytowana literatura

[1] *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Education Task Force Curriculum Committee, October 22, 2003*

---

*Andrzej Walat – wieloletni kierownik ds. nauki w OEIiZK, uznany w kraju dydaktyk informatyki z wieloletnim dorobkiem pedagogicznym w szkołach różnego typu, twórca kolejnych polskich wersji Logo dla Windows, twórca programów doskonalenia nauczycieli w zakresie edukacji informatycznej. Autor ponad 100 publikacji, w tym wielu książek i podręczników do matematyki i informatyki. Autor programów nauczania i podręczników do informatyki i technologii informacyjnej.*

---