UNIWERSYTET HUMANISTYCZNO-PRZYRODNICZY IM. JANA DŁUGOSZA W CZĘSTOCHOWIE
Edukacyjna Analiza Transakcyjna 2018, nr 7

http://dx.doi.org/10.16926/eat.2018.07.11

Zbigniew LEDÓCHOWSKI
https://orcid.org/0000-0003-0405-0817
e-mail: zbigniew.ledochowski@apsl.edu.pl
Afiliacja: Akademia Pomorska, Słupsk

# Rola sztuki programowania w kształceniu informatycznym

**Jak cytować [how to cite]:** Ledóchowski, Z. (2018). Rola sztuki programowania w kształceniu informatycznym. *Edukacyjna Analiza Transakcyjna*, *7*, 195–207.

## NAWIĄZUJĄC DO ANALIZY TRANSAKCYJNEJ (OD REDAKCJI)

Każdy fenomen społeczny, który wpływa na ludzkie zachowania, warty jest uwagi i analizy. Z punktu widzenia analizy transakcyjnej, sztuka jest fenomenem, który budzi emocje czasami trudne do wyjaśnienia. Mamy więc do czynienia ze specyficzną transakcją pomiędzy osobami lub specyficznym spojrzeniem na wytwory ludzkie. Jeśli sztukę połączy się z programowaniem, fenomen staje się dużo bardziej złożony. Piotr Grochowski (2012) podaje cechy charakterystyczne dla stereotypu programisty – informatyka: „ubierają się we flanelowe kraciaste koszule, ewentualnie swetry, noszą okulary w grubych oprawkach oraz niemodne fryzury", programista jest „całkowicie oderwany od rzeczywistości, w związku z czym nie potrafi poradzić sobie w najprostszych codziennych sytuacjach". W procesie komunikacji „nie potrafi w codziennym życiu myśleć zdroworozsądkowo, lecz w procesie postrzegania i interpretowania świata używa nieustannie kategorii i procedur wypracowanych na gruncie informatyki, a co za tym idzie, podejmowane przez niego działania są rażąco nieskuteczne bądź nielogiczne z naszego, *nieinformatycznego* punktu widzenia". Informatyk jawi się jako wieczny zablokowany Dorosły, który nie rozumie kontekstu sytuacji społecznych, przyjmuje inne niż oczekiwane sposoby rozwiązywania problemów, patrzy na świat inaczej – podobnie zresztą jak czynią to artyści. W rozwoju osobowości stan Ja-Dorosły integruje inne stany Ja i uczy się z nich korzystać. Znajdowanie piękna w linijkach kodu, czy elegancji określonego rozwiązania technicznego, można przyrównać właśnie do integracji osobowości. Zawód programisty jest historycznie czymś niezwykle nowym, obserwując zmiany w tym obszarze, mamy możliwość z perspektywy analizy transakcyjnej zobaczyć w czystej formie przekształcenia osobowości, które w innej sytuacji wymagałyby skomplikowanych procedur eksperymentalnych.

Zbigniew Wieczorek

Grochowski, P. (2012). Dlaczego śmiejemy się z informatyków? Przyczynek do kulturowej analizy współczesnych przekazów humorystycznych. *Literatura Ludowa, 3*, 47–58.

## Streszczenie

Artyzm czy rzemiosło? Jakże często to pytanie można odnieść do nauczania informatyki w szkole. Przy czym nie chodzi tylko o przykłady oczywiste, związane z tworzeniem grafiki komputerowej czy projektowaniem witryn WWW, gdzie oprócz rzemieślniczej znajomości narzędzi informatycznych pojawia się wątek kreatywności, zmysłu artystycznego ucznia. Warto przejrzeć się choćby umiejętnościom związanym z myśleniem algorytmicznym i programowaniem, bardzo ważnym wobec właśnie wprowadzanych zmian programowych w nauczaniu informatyki. Dziś programowanie, także wobec bogactwa narzędzi, ale przede wszystkim w związku z inwencją ucznia, daje pełne pole do ujawnienia niekonwencjonalnych pomysłów, oryginalnego sposobu myślenia, nieszablonowych rozwiązań. Słowem, może być sztuką wbrew niekiedy kojarzącemu się z rzemiosłem słowu kodowanie.

**Słowa kluczowe:** kształcenie, programowanie, kodowanie, edukacja informatyczna, sztuka.

## Wprowadzenie

Poszukiwanie artyzmu, piękna, sztuki w dziedzinie tak na pierwszy rzut oka technicznej (technokratycznej), jak informatyka, wydaje się być zabiegiem skazanym na porażkę. Po bliższym przyjrzeniu się różnym kontekstom tego zagadnienia (zwłaszcza edukacyjnemu, o czym głównie w tym tekście) rzecz nie jest już jednak tak (negatywnie) oczywista.

Każde rozważanie służące udowodnieniu pewnej tezy winno opierać się na założeniach. Szkopuł w tym, że nie istnieje jedna, a właściwie – zdaniem wielu – nie istnieje żadna formalna definicja sztuki. Mówimy o działalności uprawianej przez artystów (a kiedy jest się artystą?), podajemy synonimy tego określenia – jak artyzm właśnie, także piękno czy kunszt. Często przeciwstawiamy sztukę profesjonalnemu, ale jednak rzemiosłu w pewnej dziedzinie (ten wątek w zaproponowanym kontekście zresztą także się pojawi). Wśród innych prób zdefiniowania sztuki znajdujemy umiejętność, którą można realizować dzięki szczególnemu talentowi, zręczności czy kwalifikacjom, albo czyn dokonany dzięki umiejętnościom. Ta mnogość określeń sztuki, czy inaczej – brak formalnego jej zdefiniowania – paradoksalnie wyjdzie na korzyść tezie postawionej w tym tekście. Opierając się choćby na przywołanej relacji między dziełem a talentem czy umiejętnościami, spróbujemy pokazać, że pierwiastek sztuki jest jak najbardziej do odnalezienia nawet w tak hermetycznym, wydawałoby się, świecie, jak świat algorytmów i programowania. Szczególnie odniesiemy to do edukacji informatycznej, zauważając pewną szansę dla metodyki nauczania programowania w owych związkach z – być może, bardzo tu szczególnie rozumianą, ale jednak – sztuką.

Nie sposób w tym momencie nie odwołać się do bardzo skróconego rysu historycznego związanego z ewolucją informatyki, jej metod i zastosowań. Wspomnijmy tu trzy, ważne właśnie dla relacji ze sztuką, momenty. Najpierw, zaraz po powstaniu elektronicznych maszyn cyfrowych, później zwanych już komputerami, informatyka jawi się jako dziedzina jeszcze niesamodzielna, służebna in-

nym, zwłaszcza ścisłym dziedzinom wiedzy. Mówiąc krótko, komputer miał przede wszystkim ułatwić i przyspieszyć żmudne obliczenia, i tak widziano zastosowania informatyki. W tym okresie jej rozwoju raczej nie będziemy poszukiwać związków ze sztuką, niezależnie od tego, jak niemałą sztuką w bardziej dosłownym tego słowa rozumieniu było konstruowanie pierwszych komputerów. Kolejny ważny moment (trudno o precyzyjną cezurę historyczną, choćby dlatego, że w tej roli informatyka też „trwa" do dziś) to wyłonienie się informatyki jako samodzielnej dziedziny nauki, z własnymi obszarami badań i zastosowaniami raczej w dziedzinach naukowo technicznych. I wreszcie etap trzeci (utożsamiany w pierwszej fazie z pojawieniem się komputerów osobistych) – to czas, w którym zastosowania informatyki opuściły obszar naukowo-techniczny, dostarczając rozwiązań i zastosowań również biznesowi, gospodarstwom domowym, kulturze i sztuce właśnie, słowem – niemal nam wszystkim. W tej roli zresztą znaczenie informatyki odnotowujemy do dziś. Dla lepszego uwypuklenia tych zastosowań informatyki (i odróżnienia od obszarów zastosowań stricte naukowo-technicznych) stworzone zresztą zostało określenia technologii informacyjno-komunikacyjnych, czy też nowoczesnych technologii. Nie miejsce tu na rozważania językowohistoryczne, związane z ciekawą historią tworzenia się tych terminów, pewnego zamieszania, jakie na początku wprowadziły. Zauważmy, co w tym tekście ważniejsze, że właśnie z dynamicznym rozwojem nowoczesnych technologii, nowymi możliwościami, jakie stwarzają narzędzia i urządzenia, wiązać w pierwszej kolejności należy dzisiejsze oblicze informatyki ze sztuką. Dzisiaj słowo komputer jest synonimem licznych urządzeń, zwłaszcza mobilnych. Mobilność razem z multimedialnością i możliwościami komunikacyjnymi stają w szeregu terminów definiujących nowe możliwości komputerów, wychodzące daleko poza historyczne zdolności obliczeniowe (które zresztą też się wzmocniły). Z kolei te możliwości mogą wspomagać i wspierać działania twórcze, kreatywne, skąd blisko już do sztuki.

W tym tekście zechcemy jednak „poszukać sztuki" nie tam, gdzie to bardziej oczywiste, czyli w wielu dziedzinach, w których sami artyści, graficy, kompozytorzy korzystają ze zdobyczy nowych technologii. Skupimy się na bardziej „klasycznych" obszarach informatyki związanych z programowaniem i jego zastosowaniami, w dodatku odnosząc te rozważania głównie do edukacji informatycznej, co szczególnie ważne w dobie zmian programowych wprowadzających bardziej powszechnie, niż miało to miejsce dotąd, nauczanie programowania do polskich szkół. Poszukiwanie odniesień do sztuki akurat w tym kontekście nie jest na pewno tak oczywiste, jak w obszarach związanych z wykorzystaniem technologii informacyjno-komunikacyjnych, z ich coraz nowocześniejszymi rozwiązaniami, ale okaże się, że będzie to możliwe. Ten główny wątek poprzedźmy jednak najpierw ogólniejszym wstępem dotyczącym informatyki (…i sztuki) w szkole.

## Edukacja informatyczna a sztuka – uwagi ogólne

Wyżej napisano, że nowe technologie stworzyły tylko szansę dla działań twórczych, kreatywnych. Zmierzamy do tego, że sama technologia nie tworzy jeszcze automatycznie rozwiązań, w których dostrzec można artyzm i mistrzostwo. Wiele zależy przede wszystkim od ludzi, nauczycieli (Ledóchowski, 2018), uczniów, metodyki pracy z nowymi technologiami. Warto jeszcze zwrócić uwagę na jeden ważny aspekt. Pojawienie się w szkolnej rzeczywistości technologii informacyjno-komunikacyjnych (abstrahujemy tu od skali tej obecności, bo to inny ważny problem) nie oznaczało i nie może oznaczać tylko przełomu technologicznego. Bardzo istotny jest kontekst poznawczy, nazwijmy go humanistycznym. W tej materii znaczącą rolę, pomimo tego, że nie była nigdy odrębnym przedmiotem, odgrywa np. edukacja medialna. W tym kontekście popatrzmy zresztą na Internet. Od strony czysto technologicznej (szerokopasmowość), stwarza ogromne możliwości wykorzystania różnych form pracy w kształceniu i samokształceniu, ale dopiero szersze – nazwane wyżej, może nie całkiem trafnie, humanistycznym – podejście pozwala z tych form „wyciągnąć" takie czynniki, jak tworzenie i współtworzenie elementów o różnej złożoności merytorycznej i szerokim spektrum zastosowań, propagowanie rozwiązań twórczych czy choćby tzw. dobrych praktyk, wreszcie promocję swojego – instytucji – wizerunku. W odniesieniu do tematu nietrudno też przypisać Internetowi rolę swego rodzaju wirtualnej galerii sztuki, zawierającej prace zarówno na amatorskim, jak i profesjonalnym poziomie. Dodajmy, że niechcący jest też Internet elementem komercjalizacji sztuki jako takiej.

Wracając do edukacji informatycznej. Wypada tu postawić tezę, że odbieranie intelektualnego waloru i sprowadzenie do aspektów technologicznych nie da tej edukacji najmniejszych pretensji do obecności choćby pierwiastków sztuki. A obraz jest zróżnicowany. Nie ma raczej wątpliwości, że władze oświatowe doceniają rolę tej edukacji, jej cywilizacyjny wymiar. W ślad za tym idą różne działania i projekty, a od roku szkolnego 2017/2018 rozpoczęto wdrażanie zmian programowych, które mają na celu przede wszystkim upowszechnienie elementów myślenia komputacyjnego, między innymi poprzez wprowadzanie powszechnej nauki programowania. Więcej na ten temat autor pisał w 2015 roku. Z drugiej strony, jest tzw. rzeczywistość szkolna, w której kilka elementów budzi niepokój. O infrastrukturze, w wielu szkołach nieadekwatnej do nowych wyzwań, mówiono i pisano wiele. Nie wiem, czy nie ważniejszym problemem jest przygotowanie nauczycieli, albowiem ich rola w nowej rzeczywistości programowej jest kluczowa (Ledóchowski, 2018). Zauważmy dla przykładu, że obecnie, także w kształceniu podstawowym w zakresie informatyki, w szkołach średnich pojawią się elementy zaawansowane. Nie wszystkiemu organizacyjnie podołają – na ogół lepiej przygotowani – nauczyciele zajmujący się przede wszystkim kształceniem rozszerzonym w zakresie informatyki, a dla tych posiadających formalne

uprawnienia, lecz często je wykorzystujących tylko przy doraźnej potrzebie, nowe wymagania, także w kształceniu podstawowym, mogą być za wysokie w stosunku do ich, często dawno zdobytego, przygotowania i także doświadczenia. Podobny problem pojawia się w szkołach podstawowych. Najgorsze zaś, co mogłoby się wydarzyć, to zderzenie szczytnych i potrzebnych założeń programowych, entuzjazmu dzieci i ich rodziców ze swego rodzaju marazmem szkolnym, z podejściem nieprzygotowanego nauczyciela, który (żadnej dziedziny nie obrażając) nadal chciałby traktować informatykę jako przedmiot w hierarchii szkolnej drugorzędny, niemający waloru intelektualnego i stojący może nieco powyżej (też niesłusznie tak traktowanego) wychowania fizycznego. Wtedy zamiast budowania kreatywności, wspierania kunsztu i oczekiwania na „artystyczne" wytwory ze strony uczniów znowu będziemy słyszeć pejoratywne określenie o „klawiszologii" – określenie, które już i tak w przeszłości wyrządziło wystarczająco wiele złego właściwemu pojmowaniu roli edukacji informatycznej. Zarysowany obraz nie jest powszechny, w wielu miejscach mimo trudności obiektywnych nauczyciele rozumieją znaczenie edukacji informatycznej i potrafią swoją wiedzę i entuzjazm przekazać uczniom. I tam możemy oczekiwać ciekawych efektów tej pracy uczniów, a także dopatrzyć się w nich niemałego kunsztu. Właściwie realizowana edukacja informatyczna może zawierać pierwiastek sztuki, nawet w jej bardziej użytkowym sensie.

Istnieją w szkole takie obszary tematyczne (treści programowe nauczania przedmiotów informatycznych), w których narzędzia i środki informatyki (technologii informacyjno-komunikacyjnych) mogą posłużyć tworzeniu dzieł, które w dorosłym świecie zaliczają się do sztuki tworzonej z pomocą nowych technologii. Można tu wymienić – bez podziału na etapy kształcenia – tworzenie motywów graficznych (w tym grafiki trójwymiarowej), animacji, sekwencji filmowych, czy tworzenie projektów witryn internetowych. Każde z tych zagadnień daje uczniowi szansę spełnić się niejako artystycznie, ale i przed nauczycielem informatyki stawia różne nowe problemy. Chociażby kwestia oceny umiejętności uczniowskich. Wystawianie ocen dziełom artystycznym (nawet w szkolnym sensie)? Często chciałoby się powiedzieć po prostu, jakie to piękne, doceniając uczniowską kreatywność, ale wymogi szkolne sprawiają, że realizacja tego typu projektów musi być jednak oceniona. Oceniamy zatem biegłość „technologiczną", np. związaną z posługiwaniem się określonym oprogramowaniem, czy walor artystyczny? A przecież nauczyciel informatyki nie jest plastykiem czy muzykiem, ma zapewne swój gust i zapatrywania estetyczne, ale gusta mogą być tak różne. Czy jest zatem władny wytwory graficzne, muzyczne itp., stworzone przy pomocy narzędzi informatycznych, obiektywnie ocenić? Pomimo tej trudności, oceny jednak się dokonuje. Także poza szkołą. Odnotujmy istnienie dość sporej liczby konkursów, które dotyczą właśnie tworzenia grafiki, filmów czy projektów witryn, na tematy wskazane przez organizatorów. Wzbogacają one paletę konkursów *stricte* informatycznych, czyli odnoszących się do rozwiązywania

problemów związanych z algorytmiką czy analizą danych, i są całkiem chętnie wybierane przez tych uczniów, którzy – być może – w konkursach łączących poniekąd sztukę z informatyką widzą większą szansę na spełnienie się.

I jeszcze jedna uwaga natury technologicznej. Nie ulega żadnej wątpliwości, że ewolucja środowisk informatycznych (oprogramowania, i systemowego, i użytkowego) miała znaczący wpływ na to, że w ogóle możemy rozważać zagadnienie związków „twórczości" informatycznej ze sztuką. Czy można sobie wyobrazić artystów, nawet wąsko rozumianych, tworzących w trybach tekstowym czy wsadowo? Ta epoka ewolucji narzędzi informatycznych zdecydowanie artystom nie sprzyjała. Dopiero środowiska graficzne i wyspecjalizowane narzędzia (ale intuicyjne i interaktywne) zachęciły tych, którzy dotąd nie widzieli szans na pokazanie swojej kreatywności w cyfrowym świecie.

Tak jak wspominaliśmy, naszym zadaniem będzie jednak przede wszystkim poszukanie w edukacji informatycznej związków sztuki z tymi treściami programowymi, w których jej obecność jest raczej nieoczywista.

## O sztuce programowania i w programowaniu

Dlaczego właśnie o programowaniu, czy myśleniu algorytmicznym, będzie tu mowa? Odpowiedzią są zmiany programowe w polskiej szkole, o których już wyżej wspominano. W zgodzie z trendami cywilizacyjnymi także w naszych szkołach pojawia się, właśnie już od najmłodszych uczniowskich lat, nauka programowania. Naturalnie – w różnym zakresie tematycznym dla różnych etapów edukacyjnych, realizowana też za pomocą różnych środowisk zależnych od wieku ucznia, ale w miarę powszechnie. Nieco już tracący na znaczeniu termin alfabetyzacji komputerowej zostaje zastąpiony wyzwaniem na pewno większym: myśleniem komputacyjnym i programowaniem, a w związku z rzeczywistym zapotrzebowaniem na umiejętność rozwiązywania problemów z różnych dziedzin metodami znanymi z informatyki (programowania) wyzwanie to będzie dotyczyć znacznie większego kręgu uczniów niż dotychczas.

Uwzględniając nowe trendy w nauczaniu informatyki, spróbujemy w programowaniu poszukać elementów związanych ze sztuką.

Pozornie trudno znaleźć piękno lub artyzm w czynności tak „inżynierskiej", jak projektowanie rozwiązań (tworzeniu algorytmów), i zamienianiu ich na programy. Ale przywołując jedno z przytoczonych wcześniej określeń sztuki rozumianej jako czyn dokonany dzięki umiejętnościom, trudno nie zauważyć, że tworzenie programów (nawet w kontekście szkolnym) takich umiejętności bez wątpienia wymaga. Ponadto każdy program komputerowy, nawet napisany w szkole, rozwiązując konkretny problem, przynosi określony użytek. W szkole – samemu autorowi, choć niekiedy również większej grupie uczniów, którym dobre rozwiązanie może posłużyć jako drogowskaz przy pracy z kolejnymi problemami. Na-

turalnie, w przypadku programów komercyjnych, dużych systemów informatycznych krąg czerpiących z nich korzyść jest dużo większy. Można mówić zatem o swoistej sztuce użytkowej, ale w tym określeniu nie ma przecież niczego złego. Pożytek z napisanego programu dającego szansę na realizację różnych zadań może być sztuką. Piękno nie tylko zachwyca – „Bo piękno na to jest, by zachwycało do pracy" – pisał przecież Norwid.

Tworzenie programów nie jest tylko zwykłym rzemiosłem, polegającym na przetłumaczeniu algorytmu, którego konstrukcja pozwala wykazać się kreatywnością. Ten fałszywy pogląd sprostuje każdy doświadczony programista. Gdyby twórczość kończyła się na stworzeniu algorytmu, projektu, to przecież kod programu mógłby być generowany przez automat. Nie jest to do końca możliwe, a w bardziej skomplikowanych przypadkach w ogóle niemożliwe. Programista rozwiązuje wiele problemów, których nie rozważa się na etapie tworzenia algorytmu (choćby dobór struktur danych, optymalizacja kodu, itd.) – co pozwala bez żadnych zastrzeżeń jego pracę przy tworzeniu kodu programu określić jako twórczą i odróżnić od zwykłego rzemiosła. Często stworzenie optymalnego, eleganckiego kodu jest prawdziwą sztuką.

Analizując treści programowe nauczania informatyki w szkołach, warto też zauważyć, że programowanie nie musi być w edukacji kojarzone wyłącznie z umiejętnością tworzenia kodu w określonym języku. W istocie nawet używany często termin programowanie komputerów jest pewnym uproszczeniem i nie oddaje bogactwa terminu programowanie. Programujemy też inne urządzenia w życiu codziennym i w szkole (choćby roboty). Komputer jest niejako synonimem tych różnych urządzeń. Lista innych jeszcze działań ucznia wynikających z treści programowych i mogących kojarzyć się z programowaniem jest obszerna. Zaprogramowanie obrazu w edytorze graficznym lub za pomocą zorientowanego graficznie języka programowania, sterowanie robotem, tworzenie scenariusza montażu multimedialnego, tworzenie skryptów zwiększających atrakcyjną wizualność witryn WWW lub tworzących animację – te przykłady świadczą o tym, że programowanie w szkole niejedno ma imię, a uczniom daje szansę znalezienia obszarów tematycznych, w których mogą się wykazać, stać się cyfrowymi twórcami.

Ciekawym aspektem w rozważanym kontekście jest rekurencja, która sama w sobie jest metodą rozwiązywania problemów algorytmicznych, stosowaną nie tylko przy problemach graficznych. Paradoksalnie programowanie grafiki tą metodą jest doskonałym wprowadzeniem do wykorzystania rekurencji w problemach obliczeniowych, co jest demonstrowane w szkole średniej i pozwala na pełniejsze zrozumienie idei rekurencji, a przede wszystkim pozwala na trening w zakresie poszukiwania związków logicznych między elementami graficznymi, poszukiwania samopodobieństwa rodzin figur. Jednocześnie zaprogramowane rysunki zachwycają uczniów swoim pięknem i mogą poczuć się oni prawdziwymi twórcami złożonej grafiki. Mowa tu przede wszystkim o znanych (płatek Kocha, dywany Sierpińskiego) i mniej znanych fraktalach. Wiele z nich można

modyfikować, uzyskując różne efekty, nawet jeśli nie tak estetyczne, jak oryginały, to sam fakt dokonania takiej artystycznej modyfikacji pozwala jeszcze pełniej poczuć się uczniowi twórcą.

Uczniowie najbardziej uzdolnieni informatycznie interesują się programowaniem w jego – można powiedzieć – najbardziej klasycznej postaci, tzn. rozwiązują, niekiedy skomplikowane, problemy algorytmiczne oraz – z wykorzystaniem zaawansowanych technik programistycznych – tworzą, uruchamiają i testują programy. W tym przypadku sztuka rozumiana jako poziom kompetencji tworzącego programy jest, doprawdy, dość wysokiego lotu, choć nieco subtelnie ukryta, można powiedzieć, że jest to sztuka dla koneserów.

Rozwiązanie dające użytek już sugeruje paralelę ze sztuką użytkową. To jednak nie wszystko. Pomysł na rozwiązanie problemu sam w sobie może zachwycać swoją oryginalnością, pomysłowością, elastycznością. Dotyczy to przede wszystkim problemów nietypowych, bo to przy ich rozwiązywaniu ujawnić się może talent i szczególny kunszt. Zdarza się przy okazji sprawdzianów, że nauczyciel ma nieco inną wizję rozwiązania powierzonego uczniom, trudnego problemu i jest mile zaskoczony dostarczonym przez niektórych uczniów (bywa, że jednego, ale sztuka jest wszak elitarna) rozwiązaniem. Sam pomysłodawca problemu często nie dostrzegał takiej możliwości, tym bardziej docenia ukryte piękno koncepcji. W przypadku konkursów informatycznych zdarzały się sytuacje, w których organizator przygotowywał na użytek sprawdzania zadań rozwiązania zwane wzorcowymi, a potem pojawiało się rozstrzygnięcie bardziej optymalne, rozwiązujące podany problem efektywniej, i trzeba było się wycofać z terminu wzorcowe, ale także docenić kunszt tego, który rozwiązanie nadesłał. To nie są przypadki masowe, ale przecież w tym leży istota problemu – nie każdy jest artystą, zwłaszcza najwyższej klasy. Należy podkreślić, że właśnie problemy stawiane w szkole dają szansę na uzyskanie oryginalnych, pełnych kunsztu i wysokiej znajomości rzeczy rozwiązań. Nie jest tak, że i przy tworzeniu profesjonalnych systemów informatycznych nie ujawniają się genialni informatycy z własnymi pomysłami, ale w świecie tej – nazwijmy ją – profesjonalnej informatyki częściej idzie się utartymi drogami, w czym zresztą mniejsza wina samych informatyków i projektantów, a większa, jeśli tak to można określić, typowości rozwiązanych problemów, specyfiki rynku odbiorców oprogramowania, itp.

Nie tylko rozwiązanie problemu informatycznego (algorytmicznego) może zachwycać. Czasami, co docenią przede wszystkim znawcy tematu, zachwyca jego styl, sposób zakodowania, dobór struktur danych. Ten sam problem można rozwiązać i zapisać bardzo chaotycznie, ale można też w stylu, który ujmuje prostotą, przejrzystością, a czerpiąc z terminów bliskich sztuce – pięknem formy. Doceniają to zawsze nauczyciele, kiedy taki styl kodowania ułatwia im analizę czasami niebanalnego i nieoczywistego rozwiązania zadania, ale doceniają również w swoim świecie profesjonalni projektanci i programiści. Ci ostatni zwłaszcza w sytuacjach (jakże często występujących w praktyce), kiedy przychodzi im

kontynuować tworzenie kodu programu, którego pisanie rozpoczynał ktoś inny. O ile łatwiej kontynuować te pracę, kiedy wszystko to, co składa się na styl programowania ( świadomie pomijamy wszystkie składowe tego pojęcia), jest na wysokim poziomie, a ile czasu i kłopotów kosztuje próba adaptacji do kolejnych faz pracy kodu napisanego chaotycznie. Można powiedzieć, że nawet w tej specjalistycznej kwestii pojawią się artyści i zwykli rzemieślnicy, a czasami nawet chałturnicy.

W poprzedniej części zwróciliśmy uwagę, że tworzenie programu na podstawie algorytmu wymaga inwencji twórczej ze strony programisty i nie jest tylko zwykłym zakodowaniem zapisu algorytmicznego. Zatrzymajmy się chwilę nad wzajemną relacją terminów programowanie i kodowanie. Był okres, w którym terminy te wyraźnie przeciwstawiano sobie (autor tego tekstu też był zwolennikiem takiego przeciwstawienia). Mówiąc krótko, w słowie programowanie zawarta była głębsza treść niż tylko przetłumaczenie algorytmu na język zrozumiały dla komputera. Było w nim zawarte to wszystko, co decyduje o tym, że programista też musi być wysokiej klasy fachowcem, twórcą trochę innej fazy budowania systemu (programu) niż jego projektant, ale jednak twórcą. Ta twórczość wymagała często dużego kunsztu, aby uwzględnić to wszystko, co ma już program, a czego nie ma jeszcze algorytm. Kodowanie traktowano jako pojęcie pokrewne, ale w samym słowie zawarta była pośrednio idea, że pierwiastków twórczych jest tu mniej, więcej klasycznego przetłumaczenia algorytmu na program. Bardziej obrazowo, programista to było coś „wyższego" (artysta) niż – jak to mówiono niekiedy z pewną pejoratywnością – „kodowacz" (rzemieślnik?). Z perspektywy czasu tak jaskrawa różnica pomiędzy obydwoma pojęciami zaciera się. Wydaje się, że różnica tkwi raczej w złożoności problemów, które przychodzi rozwiązywać i które stwarzają szansę na zademonstrowanie większych lub mniejszych umiejętności *stricte* programistycznych. W wielu konkursach informatycznych, np. słynnej już idei popularyzującej programowanie wśród najmłodszych – Godzina Kodowania[1] – słowo kodowanie nie ma wcale sensu pejoratywnego czy umniejszającego czyjekolwiek umiejętności programistyczne. Nie przyjmując zatem, jako szczególnie istotnej, różnicy między terminami programowanie i kodowanie, na swój prywatny użytek pozostańmy z refleksją, że rozwiązując bardzo poważne problemy, wymagające stworzenia programu, można to zrobić ze swego rodzaju wdziękiem i kunsztem, albo nieco bardziej chaotycznie, w duchu, co najwyżej, dobrego rzemiosła (czyli być nie programistą-artystą, ale takim właśnie kodowaczem, w tym nieco starszym rozumieniu tego terminu).

Rozważania dotyczące oryginalności rozwiązań programistycznych, ich kunsztu, stylu zapisu dotykają niewątpliwie sfery specjalistycznej. Użyteczność stworzonego programu może docenić wielu, elementy, o których była mowa wyżej – tylko fachowcy. Ale czy to właśnie nie podsuwa bardzo wyraźnych związ-

[1] Godzinakodowania.pl

ków ze sztuką? Wielu, patrząc na obraz czy słuchając utworu muzycznego, powie, że jest piękny, ale koneser, znawca, dostrzeże coś więcej. Np. grę kolorów i motywów, lub odniesień muzycznych, zgodność z określonym gatunkiem malarskim, muzycznym, literackim – cechy, które dla zwykłego konsumenta sztuki będą niedostępne. To, co dla znawcy impresjonizmu jest wyrazem mistrzostwa artysty, dla kogoś innego może być tylko zbiorem bezładnych, kolorowych, choć urokliwych, plam na płótnie. Sztuka w pewnej części zawsze była elitarna. Nawiasem mówiąc, informatyka nie jest jedyną dziedziną, gdzie elementy, dla kogoś pozornie techniczne, fachowców będą zachwycać wewnętrznym pięknem. Tak jest również choćby w matematyce. Matematycy potrafią zachwycać się pięknem i zgrabnością danego twierdzenia oraz mistrzostwem i artyzmem dowodu. Nieprzypadkowo o wielu matematykach szkoły lwowskiej mówiło się, że formułują twierdzenia i teorie w sposób kunsztowny, wręcz artystyczny. Czytelnicy (matematycy) potrafili używać nawet określeń, że teorie te opublikowane w postaci książkowej czytało się jak najwspanialsze dzieła literackie – co oczywiście dla osób spoza kręgu zajmującego się matematyką (i to raczej wyższą) brzmi zapewne nieco dziwnie.

W tym tekście wspomniano już, że drogę do twórczości w informatyce otworzyły niewątpliwie nowe narzędzia (oprogramowanie). Dotyczy to w szczególności także narzędzi służących do programowania. Narzędzia do tworzenia programów, mowa o tych wykorzystywanych w edukacji informatycznej, są w większości wolne, co dla młodych twórców jest rzeczą bardzo istotną. Ważne, że są to często narzędzia umożliwiające pracę w środowisku graficznym, nie tekstowym (istotne dla początkujących programistów). Wśród innych cech tych środowisk programowania wymienić należy przyjazność, prostotę obsługi, możliwość użytkowania za pośrednictwem dowolnej platformy systemowej i dostępność. Wszystko to zachęca młodych ludzi do rozwijania swoich umiejętności w zakresie programowania. Niektórzy z nich stają się potem cyfrowymi twórcami w pełnym tego słowa znaczeniu. Co istotne, dzisiaj w coraz większym stopniu można programować w szkole wizualnie, a nie tylko tekstowo (nie deprecjonując roli programowania w środowiskach tekstowych, które zwłaszcza w szkole średniej pozwala na zademonstrowanie wielu istotnych zagadnień). Chodzi tu nie tylko o ułatwienia w zakresie samego procesu budowania programów, wynikające ze środowiska graficznego, ale także o to, że w sposób bardziej bezpośredni i atrakcyjny uczeń może zauważyć efekty swojej pracy – proste motywy graficzno-animacyjne, elementarne algorytmy, jak i symulacje czy większe projekty. Można zaryzykować twierdzenie, że przy środowiskach tekstowych rozwiną się i w pewien sposób pozostaną (nie zaniedbując naturalnie programowania wizualnego) przyszli programiści, projektanci, a programowanie wizualne stworzy szansę kreatywnego działania dla tych uczniów, którzy wykorzystują swój talent do zbudowania ciekawych dzieł realizujących zastosowania z innych dziedzin (co jest ważne także z punktu widzenia celów wprowadzania do szkół nauki programo-

wania). Artysta poszukuje dla siebie odpowiednich środków wyrazu, formy, dobrze zatem się stało, że dzieciom i młodzieży możemy także zaproponować szeroki wybór narzędzi do programowania – każde z nich może posłużyć do rozwoju uzdolnień lub rozwinięcia szczególnych talentów.

Na zakończenie kilka uwag o charakterze metodycznym. Metodyka nauczania programowania to jeden z głównych czynników warunkujących powodzenie śmiałych rozwiązań wprowadzających naukę programowania od pierwszych etapów edukacyjnych. Przy czym nie idzie nam o metodykę w ogóle, ale o to, co do pracy podczas zajęć z programowania mogą wnieść wskazane tu i istniejące związki edukacji informatycznej ze sztuką.

Uczeń może być twórcą także podczas zajęć z informatyki, w dodatku w szerokim zakresie dostępnych możliwości, gdyż na edukację informatyczną składa się wiele dziedzin, a i samo programowanie ma różne wątki i zastosowania. Programowania nie trzeba się zatem obawiać, ale dostosowywać do możliwości, wieku, ale i entuzjazmu uczniów. Udane pomysły wykreują swoistych artystów, nawet jeśli jedni będą nimi w większym, a inni w mniejszym stopniu. Dodatkowo nowe media, zwłaszcza Internet, dają niespotykaną szansę na promocję swojej pracy, jej rozpowszechnienie – to na ogół cieszy artystę. Uczeń programując, czy ogólnie – rozwiązując problemy przy pomocy metod i narzędzi informatyki, zgłębia otaczającą rzeczywistość, dociera do jej uroków i tajemnic, czyli jest jak twórca poszukujący.

To wszystko pozwala w intersujący sposób odwrócić problem. A może poszukiwanie piękna w rozwiązaniu użytkowym (związanym z obszarami zainteresowania ucznia), w odpowiednim doborze środków wyrazu, jest też szansą na oswojenie trudnej (jednak) sztuki programowania? Czyli od sztuki, rozumianej jako rozbudzanie inwencji, kreatywności, do tak „inżynierskiej" dziedziny, jak programowanie? Przecież to, co ścisłe, nawet inżynierskie, nie musi być nauczane w stylu technokratycznym. Wydaje się, że zwłaszcza u podstaw nauczania informatyki (podstaw programowania) droga winna prowadzić właśnie poprzez danie szansy dziecku na własną twórczość, eksperymentowanie z rozwiązaniami, nawet mimo niepowodzeń. Dopiero później można wiedzę systematyzować, wskazywać gotowe, klasyczne rozwiązania (algorytmy), uczyć ich wykorzystania w typowych i nietypowych sytuacjach. W ten sposób mamy szansę, że choć tylko niektórzy będą wysokiej klasy fachowcami, twórcami wspaniałych dzieł (informatycznych), innych zaintrygujemy w takim stopniu, że metody informatyki będą stosować w swoich dziedzinach zainteresowań, a liczba tych, których programowanie odstręczy, zupełnie będzie znikoma. Naturalnie, by tak się stało, potrzebne jest zaangazowanie, wiele zależy od przewodników, mistrzów, czyli nauczycieli. Czy wszyscy oni będą takimi mentorami, jakich często mieli dawni mistrzowie sztuki?

## Podsumowanie

Informatyka, a także programowanie, o którym głównie mowa była w tym tekście, nie musi być kojarzona z dziedziną technokratyczną, czy na użytek szkolny – z „klawiszologią". Owoce pracy uczniów na lekcji, i poza nią, mogą być wartościowymi przykładami wyrażonego talentu i kreatywności autorów – i z punktu widzenia celów samej edukacji informatycznej, i z punktu widzenia ich odkrywczości. W obu przypadkach, co staraliśmy się udowodnić, są to przykłady swoistych dzieł, nawet jeśli rzecz dotyczy sztuki czysto użytkowej i na wysoką ocenę zasługuje dopiero pod okiem specjalistów. Jak zawsze w przypadku działalności niebanalnej, będą tu lepsi, przecierający szlaki, czyli uczniowie uzdolnieni informatycznie, jak i słabsi, bo także „twórczość" informatyczna na stosownym poziomie nie jest dostępna dla każdego. Ważne, aby w ogromie dziedzin informatycznych i różnych treści programowych każdy mógł przynajmniej spróbować znaleźć miejsce dla siebie i choć w pewnym stopniu zabłysnąć jako cyfrowy twórca, na miarę swoich możliwości i talentu, wspomagany przez mądrego nauczyciela oraz przez coraz nowocześniejsze i przyjaźniejsze narzędzia informatyczne. Wyeksponowane wnioski warto jeszcze wzbogacić i tym, że spojrzenie na kształcenie informatyczne, dziś zawierające wiele nowych elementów z programowaniem na czele, może i powinno być realizowane poprzez odpowiednią metodykę, uwzględniającą indywidualność ucznia, stawiającą na spersonalizowane środowiska i warunki pracy z dziećmi i młodzieżą na lekcji informatyki. Wśród poszukiwanych rozwiązań metodycznych, umiejętność dostrzeżenia „pierwiastka sztuki" w nauczaniu programowania jawi się w świetle opisanych przykładów jako pomysł niekoniecznie ekstrawagancki, ale wręcz przeciwnie – wart uwagi, bo wprost nawiązujący do takich pojęć, jak kreatywność czy tworzenie.

Malować i czerpać z tego satysfakcję mogą wszyscy, Leonardami zostają wybrani. Nie inaczej w dziedzinie informatyki. Każdy może poszukać swojego sukcesu związanego z rozwiązaniem odpowiednich problemów informatycznych. Szkoła zaś nie powinna z pola widzenia zgubić tych najlepszych, mających predyspozycje do tworzenia w przyszłości dzieł informatycznych, z których pożytek będzie dostrzegany także w skali makro. Winna ich przygotować oraz mądrze wskazać kierunki ich dalszego rozwoju, aby taka twórczość rzeczywiście mogła mieć miejsce.

## Bibliografia

Dylak, S. (2013). *Architektura wiedzy w szkole.* Warszawa: Difin.

Ledóchowski, Z. (2015). O potrzebie powszechnego kształcenia informatycznego. W: S. Galanciak, M. Tanaś (red.), *Cyfrowa przestrzeń kształcenia, t. I* (s. 189–205). Kraków: Wydawnictwo Impuls.

Ledóchowski, Z. (2018). O roli nauczyciela informatyki w cyfrowym świecie. W: S. Galanciak, M. Tanaś (red.), *Mistrz i uczeń w cyberprzestrzeni, t. III* (s. 181–195). Kraków: Wydawnictwo Impuls.

Kalinowski, W. *Zachwyt dekretowany*. http://www.wszp.edu.pl/fileadmin/files_wszp/Instytucja_sztuki.doc, pobrane 11.10.2018.

Sysło, M.M. (1998). *Piramidy, szyszki i inne konstrukcje algorytmiczne*. Warszawa: Wydawnictwa Szkolne i Pedagogiczne. http://mmsyslo.pl/Materialy/Ksiazki-i-podreczniki/Ksiazki/Ksiazka-Piramidy-szyszki-i, pobrane 11.10.2018.

Sysło, M.M. (2014). Myślenie komputacyjne. Nowe spojrzenie na kompetencje informatyczne. W: *Materiały konferencji „Informatyka w Edukacji XXI"* (s. 15–32). Toruń: UMK.

Tatarkiewicz, W. (2005). *Dzieje sześciu pojęć*. Warszawa: Wydawnictwo Naukowe PWN.

Wing, J. (2006). Computational thinking. *Communications Of The Acm March*, *49*, No. 3. https://www.cs.cmu.edu/~15110-s13/Wing06-ct.pdf, pobrane 11.10.2018.

## The role of the art of programming IT education

### Summary

There is a longstanding debate whether computer science education is art or craft. There is no doubt that computer graphics and website design, apart from technical skills, require some degree of invention and creativity, some artistic sense. Maybe the same concerns skills related to algorithmic thinking and programming, which are very important in view of changes in the syllabus of IT teaching that are being introduced. Today, programming provides a possibility to reveal unconventional ideas, original thinking and unconventional solutions due to the wealth of tools and, above all, students' inventiveness. Thus, it may be treated as art, contrary to the widespread belief that coding involves pure craft.

**Keywords:** education, programming, coding, computer science education art.