

## Ryszard TADEUSIEWICZ

AGH, KATEDRA AUTOMATYKI

# Wszystkie barwy informatyki

Prof. dr hab. inż. Ryszard TADEUSIEWICZ

Doktor Honoris Causa dziewięciu uczelni krajowych i zagranicznych; Wiceprezes Krakowskiego Oddziału PAN, Przewodniczący Komisji Nauk Technicznych PAU; Kierownik Katedry Automatyki AGH; Przewodniczący Rady IKO WSHE; Członek Akademii Inżynerskiej, член Российской Академии Естественных Наук; Participle Pleno Jure Academiae Europensis Scientiarum Artium Litterarumque; Fellow of World Academy of Art and Science; Euroengineer FEANI; Senior Member of IEEE; professional member of ACM.



e-mail: rtad@agh.edu.pl

### Streszczenie

Przedstawiony artykuł stanowi wprowadzenie, a jednocześnie także omówienie zawartości tego numeru czasopisma „Pomiary, Automatyka, Kontrola” z uwzględnieniem faktu, że artykuły składające się na ten numer są poszerzonymi i wzbogaconymi (a także recenzowanymi!) wersjami referatów, które będą wygłaszane w ramach Czwartej Krajowej Konferencji Naukowej „Informatyka – sztuka czy rzemiosło”. Artykuły te dotyczą różnych zagadnień informatyki i jej zastosowań, jednak zebrane razem tworzą, na zasadzie synergii, pewną wartość dodaną. Wartość ta polega na pokazaniu jak bogata i jak różnorodna potrafi być dziedzina zamykana w zbiorczym terminie: Informatyka. W niniejszym artykule wprowadzającym tę różnorodność podkreślono i uwytkulono, nadając jej nazwę „wielobarwności” informatyki.

**Słowa kluczowe:** informatyka, przetwarzanie informacji, przemysłowe zastosowania komputerów.

## Varicolored computer science

### Abstract

Presented paper has two purposes. First should be treated as general introduction to the collection of papers presented on this issue of the journal. This collection is special one, because in fact the articles printed below are enhanced and reviewed versions of the presentations, which are included into program of the Fourth National Conference entitled “Computer science – art or trade”. Particular articles are related to the different aspects and different applications of computer science, but collected together in this issue can emerge some kind of additional value. This additional value depend on evidence, how reach and how miscellaneous can be computer science, nevertheless is discussed and considered under unique name. In this paper such property of computer science is discussed and stressed, under the given name “varicolored computer science”.

**Keywords:** computer science, data processing, industrial applications of the computers.

## 1. Wstęp

Zawartość tego numeru popularnego i cenionego czasopisma, „Pomiary, Automatyka, Kontrola” jest nieco nietypowa. Otóż artykuły składające się na ten numer są poszerzonymi i wzbogaconymi wersjami referatów, które będą wygłaszane w ramach Czwartej Krajowej Konferencji Naukowej „Informatyka – sztuka czy rzemiosło”. Warto podkreślić, bo ma to spore znaczenie i dla czasopisma i dla autorów zamieszczonych prac, że wszystkie te artykuły były starannie recenzowane. Niemniej fakt przeznaczenia tych prac właśnie do wygłoszenia na tej konkretnej konferencji miał silny wpływ na ich tematykę i sposób potraktowania rozważanych zagadnień. Szczegółowy przedmiot rozważań, zawartych w poszczególnych artykułach jest bardzo zróżnicowany, więc każda próba ich syntetycznego omówienia byłaby skazana na niepowodzenie. Jednak swoistym „wspólnym mianownikiem” wszystkich tych prac jest fakt zaangażowania w nich narzędzi i środków informatyki. Dodatkowym wspólnym aspektem kolekcji artykułów, który anonsuję tym artykułem wstępnym, jest fakt, że w przeważającej większości są one dziełem ludzi młodych, zbliżających się do doktoratu lub mających tę przygodę,

jaką jest zdobycie stopnia naukowego, stosunkowo niedawno za sobą.

## 2. Jedność informatyki mimo sztafety pokoleń

Jaki związek ma wiek autorów, przypomniany w ostatnim zdaniu poprzedniego podrozdziału, z treścią ich prac, a tym bardziej z treścią tego wprowadzenia?

Otóż wbrew pozorom dosyć znaczący. Informatycy mojego pokolenia swoje pierwsze prace wykonywali na maszynach lampowych (kto z młodzieży wie dzisiaj, co to znaczy?), a także funkcjonowali w warunkach niemal całkowitego braku gotowych programów użytkowych, więc absolutnie **każdą** pracę wykonywać trzeba było w ten sposób, że najpierw się samemu pisało odpowiedni program, a dopiero potem można było coś policzyć. Komputery były więc słabe (8 kilobitów uchodziło za duży zasób pamięci!), a programy z konieczności też raczej prymitywne, ale za to myśmy wiedzieli **wszystko** na temat architektury używanego komputera (bo trzeba było wiedzieć, którą przepaloną lampę wymienić, kiedy maszyna zaczynała szwankować!) a także panowali nad wszystkimi szczegółami wykorzystywanego programu.

Co więcej, zbliżając się do komputera trzeba było się **wyżyć** większości wiedzy, którą z trudem zdobyliśmy na studiach, bo na przykład biegłość w obliczaniu całek nieoznaczonych wcale nie pomagała w stosowaniu algorytmów numerycznego całkowania metodą Rungego-Kutty, a takie pojęcia jak iteracja czy rekurencja nie wchodziły w skład żadnego z wykładów i znaleźć je można było jedynie w nielicznych, trudno dostępnych książkach zagranicznych autorów. Pierwsze polskie książki dopiero powstawały i były – z dzisiejszej perspektywy – bardzo kiepskie [1].

Informatyka miała wtedy cierpki smak, bo niewiele można było tymi metodami osiągnąć (zwłaszcza w dziedzinie sztucznej inteligencji, którą się pasjonowałem od czasów studenckich), ale był to smak doprawiony dymem z ogniska pioniera wytyczającego nowe szlaki.

Dzisiejsi informatycy, których reprezentatywną próbkę stanowią Autorzy publikowanych dalej artykułów, działają w diametralnie odmiennych warunkach. Podstawowe pojęcia informatyki poznali już w szkole podstawowej (lub nawet w przedszkolu); swoją obszerną wiedzę, zdobytą na specjalistycznych studiach i licznych kursach mogą w każdej chwili wzbogacić lub zweryfikować wstukując odpowiednie hasło w Google; mają do dyspozycji komputery o szybkości działania i pojemności pamięci, które niedawno były wprost niewyobrażalne, a do każdego niemal celu mają doskonale gotowe programy, których możliwość z reguły wielokrotnie przekraczają typowe potrzeby typowego użytkownika (by wymienić tylko Matlaba [2]).

Jednak mimo tego imponującego postępu, informatyka jest w swej istocie wciąż ta sama. Dążymy w niej od lat do automatyzacji tych czynności, które człowiek realizuje dzięki posiadanym talentom i kwalifikacjom intelektualnym – i ciągle przekonujemy się, że cel, jakim jest wyposażenie maszyny w odpowiednik ludzkiego rozumu ucieka wciąż przed nami i jest wciąż tak samo odległy, jak horyzont, którego także osiągnąć ani przekroczyć nie można. Oczywiście osiągnięcia, jakie ci młodzi i zdolni prezentują [3], są nieporównywalne z tym, co myśmy uzyskiwali przed wielu laty, a granice sztucznej inteligencji przesunęły się w regiony, o których nawet marzyć nie mogliśmy budując nasze pionierskie systemy. Jednak wciąż jeszcze aktualna jest lista celów, których osiągnięcie kiedyś deklarowaliśmy w naszych programach naukowych, a które nadal osiągnięte nie zostały.

Co więcej, w szeregach tych właśnie młodych i pod każdym względem wspaniałych informatyków rodzi się wątpliwość, wyrażona lapidarnie ale bardzo trafnie w tytule wzmiankowanej wyżej Konferencji: czy to, co robią, to jeszcze jest sztuka – czy może już tylko rzemiosło? Warto się nad tym pytaniem pochylić i spróbować na nie naprawdę poważnie odpowiedzieć.

### 3. Sztuka, czy rzemiosło?

Moje pokolenie nie miało wątpliwości: to co robiliśmy w informatyce miało (według nas) wszelkie znamiona sztuki. Utwierdzały nas w tym przekonaniu dzieła klasyków, zwłaszcza Donald E. Knuth, którego pięciotomowe dzieło *The Art of Computer Programming* (TAOCP) rzeczywiście było biblią dla osób, które chcąc używać komputera skazane były na ustawiczne eksploatowanie swojej umiejętności programowania – i dlatego miały do tej czynności określony stosunek, wynikający z własnych doświadczeń.

Co powodowało, że tak zuchwale i uparcie sądziliśmy niegdyś, iż informatyka jest sztuką?

Otóż pogląd na temat artystycznej (sic!) natury programowania brał się z kilku powodów. Po pierwsze programowania niepodobna było kojarzyć z rzemiosłem, bo rzemiosło zakłada powtarzalność pewnych czynności. Rzemieślnik, zwłaszcza gdy osiągnie mistrzowską biegłość w swym zawodzie, skazany jest na auto-naśladowictwo. Szewc, który nauczył się szyć wyjątkowo wygodne i ładne buty – musi je szyć stale tak samo, bo każde odstępstwo od tego optimum oznaczałoby wyprodukowanie wyrobu **gorszego**, a więc niegodnego jego mistrzowskiej marki. Tymczasem w informatyce praktycznie każdy kolejny problem stwarzał zupełnie nowe wyzwania, wymagał odkrywania nowych reguł oraz metod (tak, właśnie **odkrywania**, a nie stosowania – bo wzorców nie było), a powtarzalność, chociaż łatwa i korzystna, mogła być stosowana w bardzo ograniczonym zakresie.

Po drugie tylko w dziedzinie sztuki może dochodzić do tego, że ten sam wyrób, spełniający te same wymagania i odpowiadający tym samym standardom, może charakteryzować się diametralnie innym wskaźnikami jakości. Chociaż trudno wymierne, istnieją kryteria, które różnicują dzieło prawdziwego artysty w stosunku do dzieła zwykłego pacykarza. Zwłaszcza w malarstwie tak bywa: ten sam fragment krajobrazu jeden namaluje tak, że ludziom dech zapiera, a inny zrobi z tego pocztówkowy widoczek. W muzyce także – jeden kompozytor ułoży z kilkuset dźwięków dzieło, którym zachwycać się będą pokolenia, a drugi napisze coś, co ludzie zapomną w godzinę po wysłuchaniu.

W informatyce też tak bywało. Mistrzowsko napisany program mógł być pod każdym względem sto razy sprawniejszy, niż program napisany poprawnie, ale właśnie bez tego artystycznego polotu – i to dawało do myślenia.

Dlatego jako pionierzy informatyki nie mieliśmy **żadnych** wątpliwości, że to, co robimy, jest rodzajem sztuki – chociaż oczywiście nie każdy z nas rodził się artystą.

Jak jest dzisiaj?

Obawiam się, że trochę inaczej.

Jednym z powodów tego stanu rzeczy jest fakt, że współczesny informatyk znacznie rzadziej tworzy coś całkiem nowego, a znacznie częściej używa narzędzi gotowych, sprawnych, udoskonalonych – ale pozbawionych tego elementu artystycznego wyzwania, który kiedyś towarzyszył kreowaniu nowych programów. To nie znaczy, że praca współczesnych informatyków jest mniej twórcza. Przeciwnie! Ale tego „czegoś”, co stygmatyzowało kiedyś informatykę jako sztukę, doszukać się dzisiaj trudniej.

Być może jest to także kwestia skali. Dawniej programy pisało się dla jednostkowych potrzeb. Na przykład uczony, który chciał odkryć nowe ciało niebieskie pisał program, który obliczał równania mechaniki nieba i pozwalał stwierdzić, że trzeba wycelować teleskop w jakiś konkretny punkt firmamentu, bo tam właśnie, w określonym miejscu bezmiernej pustki kosmosu, krąży planetoida, której nikt jeszcze nie wiedział i która czekała na swego odkrywcę. Po odkryciu i nazwaniu planetoidy program przestawał być potrzebny, a do kolejnego zadania pisało się kolejny program.

Dzisiaj wszystko stało się masowe. Programy pisze się po to, żeby je sprzedawać w możliwie dużej liczbie egzemplarzy, więc dąży się głównie do tego, żeby były mało kosztowne. O ich doskonałość nikt nie zabiega, bo niedoskonały program skłoni użytkownika do zakupu kolejnej, podobno doskonalszej wersji. Planetoidy zresztą też odkrywa się masowo – i nikogo to już dziś nie emocjonuje, nawet jeśli okazuje się, że nowa planetoida jest większa od ciała niebieskiego, które kiedyś szumnie obwołano planetą. Coś takiego się ostatnio zdarzyło z Plutonem, który długo zaszczycany był mianem 9. planety Układu Słonecznego, a potem

został „zdetronizowany” przez odkrytą 5 stycznia 2005 r. **136199 Eris**, co jednak poza wąskim gronem specjalistów i hobbystów właściwie nikogo nie zainteresowało.

Może więc tutaj tkwi odpowiedź na pytanie, czy informatyka, która kiedyś naprawdę była dziedziną sztuki – stała się już rzemiosłem? Wszak dzieło sztuki jest zjawiskiem unikatowym i jednostkowym – i to jest jego cecha niezbywalna. Chroniona pancerną płytą szklaną „Pieta” Michała Anioła jest porażająco pięknym dziełem sztuki, przed którym można stać w zachwycie godzinami. Natomiast jej kopie, sprzedawane setkami na Palcu Świętego Piotra, mają ze sztuką mniej więcej tyle wspólnego, co odlewane z gipsu krasnale ogrodowe.

### 4. Podsumowanie

Stwierdziliśmy w poprzednim podrozdziale, że różne względu spowodowały to, że informatyka, będąc kiedyś dziedziną sztuki, stała się dzisiaj rzemiosłem – i jest nim, niezależnie od tego, czy się to nam podoba, czy nie.

Można jednak zapytać, czy to dobrze, czy źle?

Pochopna odpowiedź, że to źle – byłaby chyba niezbyt trafna.

Sztuka jest z samej swojej natury elitarna. Za tym pięknie brzmiącym terminem kryje się brutalna prawda, że ze sztuką obcować może tak naprawdę niewielu ludzi. Natomiast siła informatyki i jej cywilizacyjna rola wynika z tego, że stosują ją wszyscy, wszędzie i do wszystkiego. Jeśli mówimy dziś o formowaniu Społeczeństwa Informacyjnego, to nie dlatego, że dzieła informatyki stały się jeszcze bardziej artystyczne – tylko dlatego, że stały się one **masowo** dostępne, a także dlatego, że naprawdę przydają się wielu ludziom.

Przeobrażenia, jakim podlegała informatyka w ostatnim ćwierćwieczu przypominają poniekąd przeobrażenia, jakie zaszły w sposobach komunikacji za sprawą pociągu, samochodu i samolotu. Dlatego twierdzę, że sztukę programowania porównać można do jazdy konnej. Dawniej, gdy ktoś chciał się przemieścić do innego miasta – musiał dosiadać konia i galopować wiele wiorst, żeby osiągnąć swój cel. Podobnie dawniej każdy informatyk, gdy chciał coś policzyć – musiał sobie stworzyć potrzebny do tego celu program.

Postęp cywilizacji spowodował, że dzisiaj szybciej i wygodniej możemy przemieszczać się z miejsca na miejsce samochodem, pociągiem albo samolotem, podobnie jak zadania informatyczne lepiej i skuteczniej można rozwiązywać z pomocą gotowych programów, niekiedy manifestujących moc odrzutowego Jumbo Jeta. Jazdę konną zajmują się dziś nie wszyscy, tylko wyspecjalizowani profesjonaliści: sportowcy, cyrkowcy, wołyżerowie. Tak samo programowanie stało się domeną wąskiej grupy bardzo wysoko wyspecjalizowanych profesjonalistów, zatrudnianych przez największe firmy softwareowe.

Oczywiście można uprawiać jazdę konną jako formę rozrywki albo można programować amatorsko dla przyjemności – ale to jest temat na osobne opowiadanie.

Kończąc ten artykuł chcę podkreślić, że nawet z tego skrótowego przeglądu zagadnień, wywołanego tytułem konferencji, wynika, jak bardzo różnorodna, „**wielobarwna**” potrafi być informatyka. Mam nadzieję, że to wrażenie pogłębi się po przestudiowaniu zawartych w tym numerze artykułów.

### 5. Literatura

- [1] Gościński A., Tadeusiewicz R.: Maszyny cyfrowe w sterowaniu procesów przemysłowych. Skrypt uczelniany AGH, nr 465, Kraków 1975.
- [2] Tadeusiewicz R., Ligęza A., Szymkat M. (eds.): Computer Methods and Systems. Vol. I: Plenary lectures and special session papers, Vol. II: Regu-lar sessions, ONT, Krakow, 2005
- [3] Rutkowski L., Tadeusiewicz R., Zadeh L., Zurada J. (eds.): Artificial Intel-ligence and Soft Computing – ICAISC 2006, Lecture Notes in Artificial Intelligence, Vol. 4029, Springer-Verlag, Berlin – Heidelberg – New York, 2006.