PRACE NAUKOWE Akademii im. Jana Długosza w Częstochowie
Seria: Edukacja Techniczna i Informatyczna 2006, z. I

Urszula Wlazłowska-Zając, Artur Zając
Akademia im. Jana Długosza w Częstochowie

NAUCZANIE PODSTAW PROGRAMOWANIA W UCZELNIACH WYŻSZYCH

Podczas ostatnich kilkudziesięciu lat wielokrotnie można było się przekonać o tym, jak ważne jest opanowanie podstaw sztuki programistycznej przynajmniej w jednym języku programowania i jednocześnie jak bardzo istotna jest umiejętność szybkiego i sprawnego dostosowania się do nowych aplikacji, narzędzi, języków programowania. Wiele aplikacji, narzędzi programistycznych czy języków programowania, które były popularne jeszcze dziesięć czy dwadzieścia lat temu, w chwili obecnej nie mają już żadnego zastosowania — zostały wyparte przez udoskonalone wersje lub zupełnie nowe podejścia. Ich użytkownicy jednak korzystają z nowszych wersji bez konieczności uczenia się ich na nowo. Wynika to przede wszystkim z tego, że wiele mechanizmów, podejść, które sprawdziły się w przeszłości, są stosowane nadal, mimo że już pewnie nie pamiętamy pierwszych ich zastosowań[1].

Są też takie języki programowania czy aplikacje, które sprawdziły się na tyle w pewnych obszarach zastosowań lub zostały bardzo dobrze przyjęte przez użytkowników, że uległy zakorzenieniu mimo upływu czasu i nadal są stosowane na przekór współczesnym trendom.

W związku z powyższym wydaje się uzasadnione przedstawianie podstawowych zasad programowania wspólnie dla różnych języków programowania. Istnieje bowiem wiele wspólnych elementów, które warto przedstawiać jednocześnie i zwracać od razu uwagę na te podobieństwa.

1. Wady i zalety wspólnego przedstawiania języków programowania

Nie ulega wątpliwości, że nie jest możliwe przedstawienie wszelkich niuansów, możliwości zastosowania każdego z języków programowania, gdy przedstawia się je współbieżnie. Jednak nie o to chodzi, żeby zdobyć biegłość w każdym z języków, ale o to, by poznać pewne ogólne zasady.

---

[1] A. Marciniak, *Turbo Pascal 7.0 z elementami programowania*, cz. 1, Wydawnictwo Nakom, Poznań 1994.

Wspólne ich przedstawianie zwróci uwagę uczącego się na rzeczy istotne, które z większym prawdopodobieństwem mogą być ponownie wykorzystane w nowszych językach programowania, które powstaną za piętnaście czy trzydzieści lat.

Często zdarza się, że znamy już co najmniej jeden z języków programowania, potrafimy biegle się nim posługiwać, w związku z czym nie odczuwa się motywacji do poznawania składni innych języków. Ich wyższość bowiem objawia się tylko w pewnych aspektach i nie mamy czasu ani tak silnej potrzeby wykorzystania tych dodatkowych możliwości, żeby zdobyć się na poznanie nowej składni.

Nierzadko studenci pierwszych lat studiów, na ogół kierunków informatycznych, stają właśnie przed takim dylematem. Część z nich na początku studiów poznaje na nowo język, który już bardzo dobrze znają, natomiast pozostali poznają obcy im język programowania, zaczynając naukę niemal od nowa, mimo że znają już inny język. W takiej sytuacji bardzo przydatnym może okazać się takie wprowadzenie do programowania, które uwydatnia cechy wspólne i różnice między różnymi „narzędziami", zamiast przedstawiania każdego z osobna, poczynając od początku.

Ogólna znajomość różnych składni jest bardzo przydatna na pierwszych latach studiów, kiedy koniecznym okazuje się zapoznawanie z literaturą fachową, przedstawiającą algorytmy za pomocą, czasami już nie używanych tak powszechnie języków.

2. Wspólne cechy podejść do wprowadzenia do programowania

Nie ulega chyba wątpliwości, że najważniejszą cechą wspólną są same algorytmy, które są niezależne od języków programowania. Każdy algorytm można przedstawić w sposób zupełnie niezwiązany z jakąkolwiek składnią.

W obrębie samej składni języków ważne jest zwrócenie szczególnej uwagi na takie elementy, jak:

— podstawowe typy danych;
— zmienne;
— wyrażenia;
— instrukcje;
— funkcje, procedury.

Bazując na takim wprowadzeniu, można już zaprezentować podstawowe algorytmy i struktury danych, a co najważniejsze, w sposób dość niezależny od składni konkretnego języka programowania. Można także pokazać słabe punkty pewnych rozwiązań w jednych językach oraz zalety innych.

Omawianie współczesnych języków programowania nie może oczywiście opierać się wyłącznie na programowaniu strukturalnym. Także podejście obiektowe może być przedstawione wspólnie dla kilku języków programowania[2]. Punktami wspólnymi dla takiego omówienia będą zapewne następujące elementy:

— klasy;
— obiekty;
— dziedziczenie;
— polimorfizm.

[2] B. Stroustrup, *Język C++*, Wydawnictwa Naukowo-Techniczne, Warszawa 1997.

Istotnym może okazać się rozpoczęcie wprowadzenia do programowania obiektowego, poczynając od przedstawienia podstaw modelowania obiektowego na przykładzie chociażby języka UML[3].

3. Podsumowanie

Zaprezentowane podejście wydaje się znacznie bardziej przydatne dla studentów różnych kierunków studiów, dla których umiejętność posługiwania się językami programowania stanowi narzędzie ułatwiające prowadzenie badań, a nie cel sam w sobie. Z pewnością konieczne będzie później głębsze poznanie co najmniej jednego z języków, ale będzie to znacznie łatwiejsze i równie proste będzie przekwalifikowanie się w razie potrzeby wykorzystania innej aplikacji, narzędzia programistycznego czy języka programowania.

Ważnym elementem powyższego podejścia jest selekcja przedstawianego materiału. Koniecznością jest porzucenie omawiania pewnych aspektów programowania w konkretnych językach. Nie wszystkie bowiem informacje są równie ważne i nie wszystkie są wspólne dla większości języków.

Urszula Wlazłowska-Zając, Artur Zając

TEACHING OF ELEMENTS OF PROGRAMMING IN HIGH SCHOOLS

Summary

This article presents a new approach to teaching programming languages in polish schools. It describes a method which main aspect is in showing differences between some language syntaxes. It isn't possible to explain all the nuances of programming techniques but the most common can be presented at school level.

[3] G. Booch, J. Rumbaugh, I. Jacobson, *UML. przewodnik użytkownika*, Wydawnictwa Naukowo-Techniczne, Warszawa 2001.