

Warszawa, 27 IV 2011

Memorandum w sprawie utrzymania przedmiotu Informatyka - programowanie
w programach nauczania na kierunku Budownictwo

Informatyka jest dyscypliną naukową i techniczną, która zajmuje się przede wszystkim technologiami tworzenia systemów przetwarzających informacje. Jest ona także integralną częścią współczesnej matematyki, która dostarcza jej niezbędnych podstaw teoretycznych. Współczesna wiedza o budownictwie lądowym jest formułowana, rozwijana i prezentowana, w postaci różnego typu algorytmów znajdujących zastosowanie począwszy od inżynierii produkcji, komunikacji, zarządzania i materiałów, a skończywszy na mechanice konstrukcji, teorii sprężystości i plastyczności oraz metodach numerycznych służących przeprowadzaniu inżynierskich obliczeń. Pomijając klasyczne algorytmy, na przykład typu dziel i zwyciężaj, min-max oraz sortowania wykorzystywane między innymi w teorii szeregowania zadań procesów produkcyjnych, w planowaniu komunikacji, w prowadzeniu prac budowlanych, we wstępnych etapach projektowania kształtu budynków uwzględniających warunki zabudowy, a także bardzo zmatematyzowane algorytmy wykorzystywane m.in. w analizie numerycznej statyki i dynamiki konstrukcji budowlanych, pojawiają się one w bardzo wielu innych dziedzinach jak i przedmiotach nauczania na Wydziale Inżynierii Lądowej.

Przykładowo, w pakiecie Office (Word, Excel, Access) powszechnie wykorzystywane są (jednak tylko przez osoby, które to potrafią) możliwości jakie daje oprogramowanie tzw. makr, które w znaczny sposób ułatwiają pracę w tym pakiecie. Także coraz częstsze jest implementowanie różnego typu algorytmów przez inżynierów projektantów. Przykładowo, systemy graficzne takie jak Auto-CAD, Microstation oferują, oprócz podstawowych narzędzi dostępnych jako rozkazy standardowego menu, także inne sposoby pracy w tych systemach w postaci wbudowanych języków programowania, które w znaczny sposób ułatwiają pracę. Istnieje wiele tzw. programów-nakładek na te programy (BeStCad, RCAD, itp.) napisanych z myślą o inżynierach budowlanych, i które coraz częściej tworzone są przez samych inżynierów budowlanych posiadających odpowiednie doświadczenie związane ze sporządzaniem dokumentacji budowlanej oraz odpowiednią wiedzę na temat programowania. Współczesne programy do grafiki wektorowej mają wbudowane możliwości pisania makr przez użytkowników, pod warunkiem jednak, że użytkownik potrafi samodzielnie napisać taki program (liczący z reguły nie wiele ponad kilkadziesiąt linii kodu). Istnieje klasa programów do obliczeń inżynierskich (często przeznaczona dla inżynierów budowlanych – np. SOFiSTiK), których obsługa wymaga tworzenia tekstowych plików wsadowych lub też pożądana jest umiejętność obsługi programu umożliwiająca znacznie lepszą kontrolę nad analizowanym problemem. Pliki wsadowe przygotowuje się w jednym z języków uniwersalnych (np. C, Basic, Python) lub w językach przeznaczonych wyłącznie do pracy z danym programem (np. CADINP). Wszystkie te języki posiadają wspólne narzędzia, takie jak pętle, instrukcje warunkowe, praca z tablicami, możliwość pracy z innymi plikami, itp.. W przypadku braku lub bardzo małej znajomości programowania w którymkolwiek języku istnieje bardzo duża bariera przy nauce i użytkowaniu tego typu programów, będących w powszechnym użyciu.

Pomimo coraz powszechniejszego zastosowania komputerów w życiu codziennym w dalszym ciągu zdecydowana większość studentów pierwszego roku nie miała do czynienia z programowaniem lub miała z nim styczność w sposób bardzo ograniczony. Nie jest zauważalna zmiana jakości w prowadzeniu przedmiotów informatycznych w szkole średniej w ciągu ostatnich lat lub też zmiana ta postępuje w sposób powolny. Istnieje bardzo duża rozbieżność w poziomie wiedzy z informatyki u studentów pierwszego roku, kiedy to część z nich posiada przynajmniej podstawową umiejętność programowania w którymś języku wysokiego poziomu, a część ma problemy z podstawową obsługą komputera. Obowiązkiem uczelni technicznej wynikającym choćby z faktu, że nie wszyscy maturzyści mieli w szkołach średnich przedmioty informatyczne lub uczestniczyli w tego typu zajęciach, ale o znacznie okrojonym zakresie programowym, jest danie przynajmniej szansy na wyrównanie poziomu wiedzy informatycznej dla wszystkich studentów. Wielu studentów docenia oferowaną przez uczelnię taką ofertę i przykłada się do nadrobienia zaległości. Część zaczyna zdawać sobie sprawę zbyt późno i żałuje, że nie przykładła się zbyt intensywnie do nauki programowania, co daje często wyraz w ankietach semestralnych. Od kilku lat, daje się także zauważyć wzrastającą liczbę studentów, dla których przedmiot Informatyka1 zawiera zbyt mało informacji i deklaruje, że woleliby uczestniczyć

w znacznie rozszerzonym pod względem treści i godzin kursie programowania. Studenci, którzy otwarcie krytykują (przynajmniej w ankietach) przedmiot, z reguły mają problemy z innymi, równolegle prowadzonymi przedmiotami i ich opinia o przedmiocie pod tym względem nie może być uznana za w pełni obiektywną. Niezależnie od opinii studentów, także opinie bardzo wielu absolwentów naszego wydziału wskazują wyraźnie, że umiejętność formułowania zadań w postaci algorytmów, a także ich implementacji w jednym ze współcześnie znanych językach programowania wysokiego poziomu, jest bardzo potrzebna, przydaje się w bardzo wielu sytuacjach, które często nieoczekiwanie pojawiają się w pracy zawodowej i pomaga rozwiązać wiele problemów zawodowych, które bez posiadania umiejętności pisania choćby najprostszych programów, nie mogłyby być samodzielnie rozwiązane.

W ostatnim dziesięcioleciu, ugruntowała się w środowisku inżynierskim pewna błędna opinia, że systemy obliczeniowe, takie jak np. ABC PŁYTA, RM-WIN, ROBOT, ABAQUS, ANSYS, LS-DYNA, stawiają użytkownikowi wymagania jedynie poprawnego wczytywania danych i poprawnego interpretowania wyników. Jest to jednak bardzo uproszczony i nieprawdziwy obraz traktowania tego typu systemów wyłącznie jako czarnych skrzynek, które obliczają, wymiarują, projektują i sporządzają ostateczne wizualizacje gotowych projektów. W każdym z wyżej wymienionych czy podobnych systemach, powszechne stało się już od wielu lat udostępnianie użytkownikowi narzędzi programistycznych, opartych o standardowe języki wysokiego poziomu, które umożliwiają nie tylko znacznie pełniejsze wykorzystanie mocy tych systemów, ale przeprowadzenie obliczeń dla bardzo specyficznych i nieujętych w tradycyjnie rozumianych poleceniach interfejsu graficznego użytkownika zadań. Bez podstaw znajomości pisania prostych programów, narzędzia te nie mogą być w oczywisty sposób używane bez dodatkowych kursów jak i bez sporych nakładów finansowych i czasowych na samodzielne doksztalcanie.

Nie należy także zapominać, że absolwenci naszego Wydziału kończąc studia i broniąc dyplom w wybranej przez siebie specjalności, powinni być przygotowani do pracy zawodowej również i w innych przedmiotach – niekoniecznie odpowiadających wybranej specjalności. Mogą na przykład znaleźć zatrudnienie w firmach, w których tworzone jest oprogramowanie komercyjne dla szeroko rozumianego budownictwa. Zatrudnianie młodych inżynierów, znających

przynajmniej w stopniu podstawowym jakiś język programowania i posiadających umiejętności formułowania zadań algorytmicznych do zespołów programistycznych pracujących nad modyfikacjami, poprawianiem i ulepszaniem aktualnych systemów obliczeniowych, a także wdrażaniem nowych systemów nie tylko jest korzystne dla firmy i absolwenta, ale jest także świetną reklamą Wydziału i Uczelni. Kilkadziesiąt lat temu, kiedy pojawiały się pierwsze komputery, przedmioty informatyczne były traktowane przez całe środowisko akademickie bardzo poważnie o czym między innymi świadczyła liczba godzin ćwiczeń i wykładów. Wielu absolwentów tamtych roczników zajmuje dzisiaj eksponowane stanowiska nie tylko w nauce Polskiej ale i w bardzo wielu firmach, w tym zagranicznych, wytwarzających komercyjne oprogramowanie inżynierskie (np. w firmie ABAQUS – światowego lidera tego typu oprogramowania). Było to możliwe dzięki temu, że powszechnie już wtedy było nauczanie informatyki. Po latach, wielu z tych właśnie pierwszych adeptów inżynierów-programistów stworzyło w Polsce i na świecie powszechnie dziś używane systemy obliczeniowe. Pozbawianie szansy wielu zdolnym studentom nauczania się na wstępnym etapie studiów inżynierskich podstaw formułowania zadań w postaci algorytmów i późniejszej ich implementacji w dowolnym języku programowania, poprzez dość znaczne ograniczanie liczby godzin tego przedmiotu nie może być traktowane jako kontynuacja dość powszechnie przyjętej na uczelniach technicznych w ubiegłych latach strategii podwyższania kwalifikacji absolwentów w zakresie szeroko rozumianej kultury informatycznej.

Brak umiejętności algorytmicznego myślenia jest w spektakularny sposób widoczny na przykładzie poziomu prezentowanej w Euro-Normach wykładni projektowania konstrukcji żelbetonowych, stalowych i bardzo wielu innych. Powszechnie znana jest bardzo krytyczna opinia środowisk inżynierów budownictwa o braku przejrzystości i klarowności przekazywanych treści, które są przecież zbiorem reguł jakie powinny być bezwzględnie przestrzegane w procesie projektowania konstrukcji. W porównaniu z ciągle jeszcze (na szczęście) obowiązującymi normami polskimi, treści zawarte w Euro-Normach są wręcz akademickim przykładem braku umiejętności precyzyjnego formułowania algorytmów i procedur dla inżyniera projektanta, który nie powinien mieć tak wielkich problemów z interpretacją tekstu i co gorsza ze sprzecznościami jak ma obecnie przy próbach ich stosowania.

Opinia na temat przydatności nauki programowania na kierunku Budownictwo jest powszechna. Wyrażają ją także absolwenci naszego Wydziału – często projektanci z uprawnieniami z co najmniej kilkuletnim stażem pracy w zawodzie. Wiedza nabyta podczas studiów z zakresu nauki podstaw programowania przydaje się w pracy inżyniera budowlanego. Często w sposób istotny ułatwia pracę i sprawia, że niekiedy proste problemy nie stwarzają bariery, której pokonanie przez osobę nie znającą podstaw programowania jest trudne. Daje się to zauważyć w pracy inżynierów, kiedy to osobom obeznanym z programowaniem rozwiązanie niektórych problemów inżynierskich nie stwarza większego kłopotu, a dla osób bez takiej znajomości jest to problem często nie do rozwiązania.

Jednym z punktów wyróżniających nasz Wydział na tle niektórych innych wydziałów budowlanych jest nauka programowania na wczesnym etapie studiów. Pozwala ona na lepsze wykorzystanie narzędzi używanych w późniejszym okresie w pracy inżyniera, począwszy od programów biurowych typu Office, arkuszy kalkulacyjnych (np. MathCad), programów do grafiki wektorowej (np. AutoCAD), programów do obliczeń (SOFiSTiK, ABAQUS, itp.) i innych. W każdym z nich istnieje możliwość implementacji wiedzy zdobytej na zajęciach z Inf.1 lub z Inf.2, a tym samym bardziej optymalnej pracy z tymi programami. Wielu absolwentów naszego wydziału wykorzystuje te umiejętności w swoim życiu zawodowym.

Celem przedmiotu Informatyka1 prowadzonego na II semestrze studiów stacjonarnych oraz Informatyka2 prowadzonego na IV semestrze studiów niestacjonarnych jest prezentacja i nauka wielu, bardzo różnorodnych algorytmów m.in. algorytmu Euklidesa znajdowania największego wspólnego dzielnika dwóch liczb naturalnych wraz z matematycznym dowodem jego poprawności, algorytmów: sumowania szeregów skończonych i nieskończonych, sortowania, rekurencyjnych, algebry macierzowej, znajdowania sił w kratownicach statycznie wyznaczalnych (według Prof. Z. Kączkowskiego), Runge-Kutta numerycznego rozwiązywania układu równań różniczkowych zwyczajnych i wielu innych. Implementacja tych algorytmów przeprowadzana jest w języku C++ i VBA, których semantyka jest bardzo podobna do wszystkich innych znanych i popularnych języków programowania. Kwalifikacje zespołu prowadzącego zajęcia z przedmiotów Inf.1, Inf.2 pozwalają na wprowadzenie szeregu zmian, szczególnie w kwestii doboru przykładów. Bardzo ograniczona czasowo liczba godzin, stanowi jednak barierę w rozszerzeniu programu, ponieważ prowadzący zajęcia stawiają czoła dwóm podstawowym wyzwaniom: nauczenie ze zrozumieniem oraz nauczenie podstaw programowania w określonym języku, przy założeniu, że słuchacze stykają się z programowaniem po raz pierwszy w życiu (założenie prawdziwe w stosunku do większości studentów). Mają na to, studenci i wykładowcy, tylko 30 godzin zajęć – bez wykładu.

Prof. dr hab. inż. Tomasz Lewiński
Kierownik Katedry
Mechaniki Budowli i Zastosowań Informatyki

Dr inż. Sławomir Czarnecki,
Dr inż. Tomasz Łukasiak
Odpowiedzialni za przedmioty
Informatyka 1 (st. stacj.)
Informatyka 2 (st. niestacj.)