

Agnieszka Borowiecka

Uczymy (się) zdalnie programowania¹

Już w latach siedemdziesiątych XX wieku w wybranych liceach pojawił się przedmiot *podstawy informatyki*, a w latach osiemdziesiątych pierwsze mikrokomputery trafiły do szkół. Reforma systemu oświaty i zmiany w świecie komputerów (pojawienie się wizualnej wersji systemu operacyjnego, powszechność dostępu do Internetu, mnogość aplikacji użytkowych i edukacyjnych) spowodowały ewolucję przedmiotów informatycznych w szkole, począwszy od *elementów informatyki*, poprzez *informatykę* i *technologię informacyjną*, aż do obowiązujących dzisiaj *zajęć komputerowych* w szkołach podstawowych oraz *informatyki* w gimnazjach i szkołach ponadgimnazjalnych. Początkowo, ze względu na stosunkowo niewielką liczbę dostępnych dla wszystkich programów użytkowych oraz edukacyjnych, lekcje związane z komputerami dotyczyły przede wszystkim zarządzania systemem operacyjnym i programowania. Później dominowały zajęcia związane z wykorzystaniem technologii informacyjno-komunikacyjnych, a coraz mniej było lekcji programowania i algorytmiki – tematy te pojawiały się praktycznie jedynie na rozszerzonej informatyce w szkołach ponadgimnazjalnych.

Od roku szkolnego 2009/2010 wprowadzana jest nowa podstawa programowa kształcenia ogólnego i nowe podręczniki. Ponownie zwrócono uwagę na znaczenie znajomości algorytmów i umiejętności pisania programów. Nowa podstawa programowa dla gimnazjum i szkoły ponadgimnazjalnej zawiera zapisy dotyczące stosowania podejścia algorytmicznego do rozwiązywania problemów z wykorzystaniem komputera oraz użyciem języka programowania. Na drugim etapie edukacyjnym (klasy IV-VI szkoły podstawowej) pojawiają się zapisy: *uczeń za pomocą ciągu poleceń tworzy proste motywy lub steruje*

obiektem na ekranie. Reforma zakłada, że każdy młody Polak po ukończeniu szkoły powinien posiadać umiejętność rozwiązywania problemów algorytmicznych i pisania programów komputerowych przynajmniej w podstawowym zakresie. Tymczasem według badań Eurostatu umiejętność programowania w grupie wiekowej 16-24 lata posiada w Polsce 16% populacji przy średniej w UE wynoszącej 20%. Przed nauczycielami stoi zatem gigantyczne zadanie – przygotować siebie i swoich uczniów do nauki programowania.

Ułatwienie może stanowić fakt, że praktycznie każdy z naszych uczniów potrafi dziś posługiwać się urządzeniami mobilnymi i różnorodnymi narzędziami internetowymi. Spróbujmy zatem wykorzystać tę umiejętność do nauki programowania – użyjmy zasobów sieciowych.

Nauczycielu, ucz się sam

Zanim zaczniemy lekcje programowania, powinniśmy sami uzupełnić swoją wiedzę. Możemy sięgnąć do różnych podręczników do informatyki, ale niestety często nie powstały jeszcze polskie wydania książek dotyczących najnowszych wersji programów, z których chcemy korzystać na lekcjach. Istniejące podręczniki nie zawsze odpowiadają naszym potrzebom lub są po prostu bardzo drogie. W sieci możemy znaleźć wiele bezpłatnych książek dotyczących programowania, także w języku polskim, np. jeśli chcemy nauczyć się programować w języku Python, możemy zapoznać się z polskim podręcznikiem powstającym na podstawie książki „Dive into Python” Marka Pilgrima, dostępnym na stronie http://pl.wikibooks.org/wiki/Zanurkuj_w_Pythonie.

¹ Fragmenty tego artykułu były drukowane w EduFakty Uczę Nowocześnie nr 20/2012.

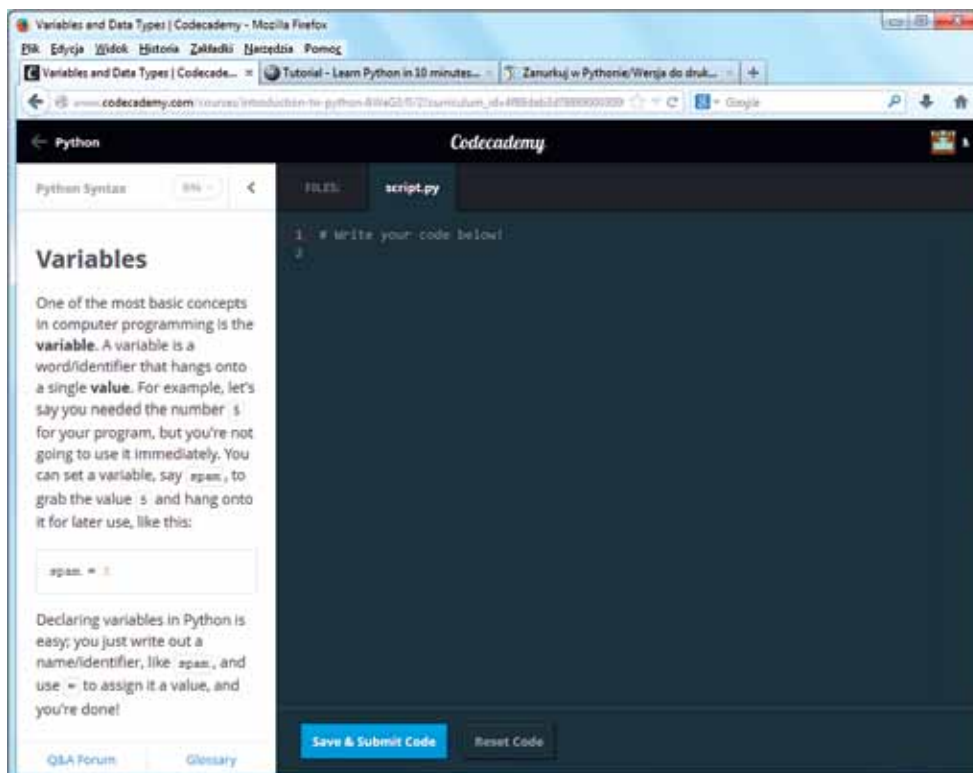
Jeśli książka to dla nas za mało, możemy poszukać w sieci interaktywnych kursów, jednak jest to trudniejsze zadanie. Materiały w postaci plików pdf lub html znajdziemy bez problemu – także w języku polskim. Wystarczy w wyszukiwarce Google wpisać np. „kurs Python”. Możemy także znaleźć różnej jakości filmy wideo w serwisie YouTube, dotyczące programowania, wpisując np. „video kurs Python”.

Interaktywną naukę programowania oferuje portal Codecademy. Obecnie dostępne są tu kursy języków JavaScript, HTML/CSS, PHP, Python, Ruby oraz Application Programming Interfaces (APIs). Po wybraniu interesującego nas kursu zostajemy przekierowani do okna z materiałami. Po lewej stronie wyświetlany jest krótki opis zagadnienia oraz ćwiczenie do wykonania. Większość okna zajmuje edytor kodu, w którym wpisujemy tworzone przez siebie programy. Gotowe rozwiązanie przesyłamy przez naciśnięcie przycisku **Save & Submit Code**. Jeśli zadanie wykonaliśmy poprawnie, pojawi się przycisk **Start Next Lesson** umożliwiający przejście do kolejnej lekcji. Za poprawne rozwiązania dostajemy także punkty i *badge* (znaczkę, plakietkę) motywujące do dalszej pracy. Codecademy umożliwia także tworzenie własnych kursów.

Jeśli interaktywność nam nie wystarczy, do nauki potrzebujemy bowiem kontaktu z innymi

uczestnikami szkolenia, warto zainteresować się platformami takimi jak Coursera czy Udacity. Oferowane przez nie szkolenia są monitorowane przez prowadzących, a ich uczestnicy często wymieniają się wiedzą i służą sobie nawzajem pomocą. Każdy kurs nie tylko wzbogaci naszą wiedzę, ale jest również źródłem wielu ciekawych pomysłów na lekcje oraz doskonałym przykładem rozwiązań metodycznych. Szczególnie polecamy kurs **An Introduction to Interactive Programming in Python** na platformie Coursera. Jest on bardzo dobrze przemyślany oraz przygotowany zarówno metodycznie, jak i technicznie, a jego tematyka – tworzenie prostych gier w języku Python – na pewno zainteresuje naszych uczniów. Jedyną wadą szkoleń na platformach Udacity i Coursera jest to, iż są dostępne głównie w języku angielskim (Coursera oferuje również kursy w językach hiszpańskim, francuskim, niemieckim, włoskim czy chińskim, ale jest ich stosunkowo niewiele). Jednak ta pozorną wadą może okazać się zaleta – znajomość języka obcego jest w dzisiejszym świecie absolutnie niezbędna, a poznanie sformułowań dotyczących programowania może się okazać przydatne w pracy. Kursy typu MOOC (Massive Open Online Course) mogą nam także pomóc nawiązywać kontakty z uczestnikami szkolenia pochodzącymi z całego świata.

Szkolenia *online* dotyczące programowania oferowane są także w Polsce przez uczelnie wyższe:



Rys. 1. Kurs Pythona na Codecademy

Uniwersytet Gdański – kurs **C#.NET**, Polski Uniwersytet Wirtualny – **Programowanie gier komputerowych**, **Programowanie internetowe**, **Programowanie obiektowe** oraz placówki doskonalenia nauczycieli: Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie – **Podstawy programowania w języku Java**, **Programowanie w języku PHP**, **Programowanie w Logo**, **Wprowadzenie do programowania w języku C++**. Część takich szkoleń jest dofinansowywana z różnych źródeł, np. grantów kuratorskich lub funduszy europejskich, i dzięki temu jest bezpłatna lub stosunkowo niedroga (około 100-200 zł). Uczestnicząc w nich, możemy łatwiej uzyskać rozwiązania konkretnych problemów, z jakimi spotykamy się na lekcjach, ponieważ nie ogranicza nas bariera językowa, a grupy są zwykle mniej liczne i prowadzący szkolenie może nam poświęcić więcej uwagi.

MOOC – kursy dla każdego

Massive Open Online Course (MOOC) to nowy rodzaj kursów *online*, zakładających szeroką skalę interaktywnego uczestnictwa i otwarty dostęp za pośrednictwem Internetu. Oprócz tradycyjnych materiałów szkoleniowych, takich jak filmy, lektury i zestawy zadań, kursy MOOC zapewniają interaktywne forum użytkowników, pomagające budować wspólnotę studentów, profesorów i asystentów. Pierwsze kursy tego typu zaczęły się pojawiać już w 2003 roku w Wielkiej Brytanii, ale ograniczały się do kilkudziesięciu studentów. Kolejne próby miały miejsce w 2007 roku w Irlandii i w 2008 roku w Kanadzie. Termin MOOC został użyty po raz pierwszy w 2008 roku przez Dave'a Cormiera z University of Prince Edward Island.

„The New York Times” nazwał rok 2012 „Rokiem MOOC”. Związane to było przede wszystkim z pojawieniem się kilku platform, takich jak Udacity, Coursera i edX, współpracujących z najlepszymi uniwersytetami z całego świata. Jesienią 2011 roku Uniwersytet Stanford uruchomił trzy kursy *online*. Wśród nich znajdował się kurs **Introduction Into AI** (Wprowadzenie do sztucznej inteligencji) Sebastiana Thruna i Petera Norviga, na który zapisało się blisko 160 000 studentów. Kolejne szkolenie, **Machine Learning**, które prowadzili Daphne Koller i Andrew Ng, zgromadziło 100 000 studentów. Sebastian Thrun jest jednym z założycieli Udacity, zaś Daphne Koller i Andrew Ng są współzałożycielami platformy Coursera.

Platforma Udacity proponuje szkolenia na trzech poziomach trudności: dla początkujących, średnio zaawansowanych i zaawansowanych. Zanim zapiszemy się na wybrany kurs, możemy obejrzeć krótkie wideo przedstawiające główne założenia szkolenia, zapoznać się z wymaganiami, sylabusem szkolenia oraz notką o prowadzących zajęcia.

Większość kursów proponowanych przez Udacity nie ma żadnych ograniczeń czasowych – możemy rozpocząć je w dowolnej chwili, lekcje są dostępne w systemie 7 × 24 (siedem dni w tygodniu, 24 godziny na dobę). Wśród szkoleń dotyczących programowania, w sierpniu 2013 znalazły się dwa szkolenia dla początkujących:

- **Introduction to Computer Science** – prowadzony przez Caya Horstmann (San Jose State University), prezentujący podstawy programowania zorientowanego obiektowo w języku Java,



Rys. 2. Strona główna serwisu Udacity oraz wizytówka przykładowego kursu

- **Introduction to Programming** – prowadzony przez Davida Evansa (University of Virginia), omawiający tworzenie programów w języku Python na przykładzie wyszukiwarki internetowej.

Szkolenia te zorganizowane są w podobny sposób. Materiał został podzielony na jednostki lekcyjne, każda lekcja składa się z kilku lub kilkunastu części – krótkich filmów wideo przerywanych pytaniami testowymi sprawdzającymi przyswajaną wiedzę. Część quizów polega na wybraniu odpowiedniej opcji lub wpisaniu prawidłowej odpowiedzi, inne wymagają napisania i uruchomienia prostego programu. Udacity zostało wyposażone w prosty edytor *online*, a testowanie działania programów jest możliwe po naciśnięciu przycisku **Test Run**, zatem uczestnik

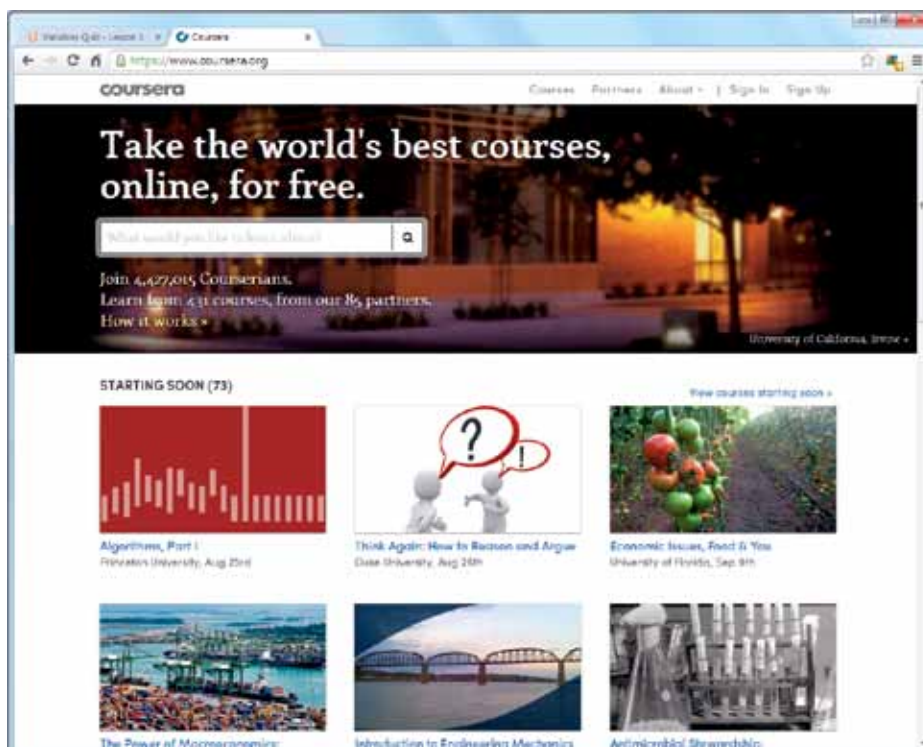
kursu nie musi mieć dostępu do komputera z zainstalowanym dodatkowym oprogramowaniem. Po zatwierdzeniu odpowiedzi wyświetlany jest kolejny fragment wykładu, zawierający omówienie prawidłowego rozwiązania. Po każdej lekcji następuje właściwy quiz sprawdzający wiedzę uczestnika. Filmy wykorzystywane przez portal są przechowywane w serwisie YouTube, ale można je także pobrać na własny komputer.

Udacity oferuje możliwość komentowania poszczególnych fragmentów materiałów zarówno przez autora szkolenia (**Instructor Notes**), jak i jego uczestników (**Discussion**).

Na platformie Coursera znajdziemy ponad 400 kursów zgrupowanych w 26 kategoriach te-



Rys. 3. Przykładowa lekcja i okno do edycji kodu na kursie Introduction to Computer Science



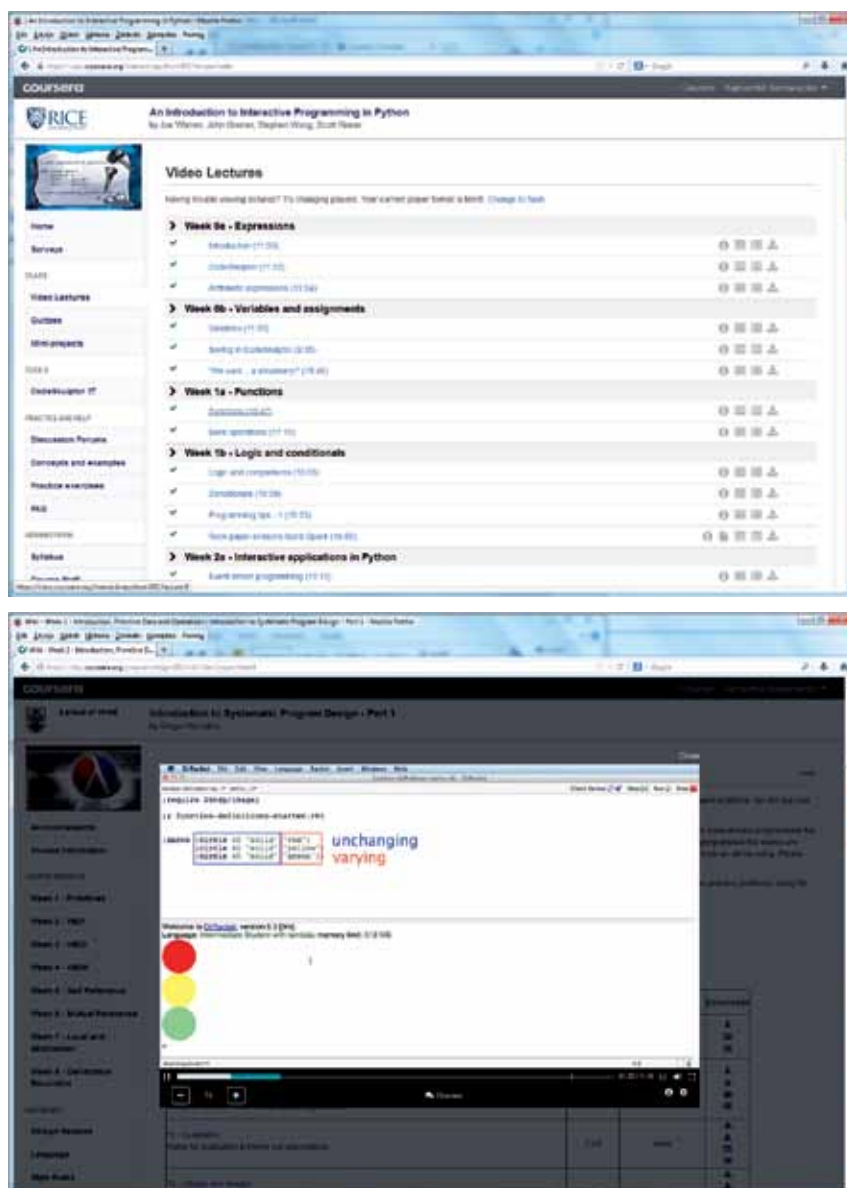
Rys. 4. Strona startowa portalu Coursera

matycznych. Szkolenia dotyczące komputerów są dostępne w czterech kategoriach: CS: Artificial Intelligence, CS: Software Engineering, CS: Systems & Security oraz CS: Theory. Dla nauczycieli interesujące mogą być również szkolenia dotyczące edukacji (kategoria Education). Coursera organizuje czas szkolenia inaczej niż Udacity. Co prawda materiały są również dostępne w systemie 7×24 , ale szkolenia rozpoczynają się konkretnego dnia, trwają określoną liczbę tygodni, a przy rozwiązywaniu zadań należy przestrzegać terminów – inaczej utracimy pewną liczbę punktów. Szkolenia, które cieszyły się dużym zainteresowaniem, są cyklicznie powtarzane i udoskonalane.

Podobnie jak na Udacity, przed zapisaniem się na szkolenie na platformie Coursera możemy zapoznać się z krótkimi informacjami na jego

temat: opisem, wymaganiami wstępnymi, wizytówkami prowadzących oraz filmem wideo wprowadzającym w tematykę kursu. Po rozpoczęciu szkolenia możemy zapoznać się ze stopniowo udostępnianymi materiałami. W menu po lewej stronie okna kursu wyświetlane są linki do informacji o kursie, ankiet, wideowykładów, quizów itp. Rodzaj i kolejność odnośników zależy od konkretnego szkolenia. Materiały można oglądać *online* lub pobrać na własny komputer.

Wykłady są dzielone na kilka części, długości poszczególnych filmów są różne – zwykle od kilku do kilkunastu minut, choć zdarzają się również 25-minutowe. Coursera oferuje narzędzie pozwalające dodać w środku lub pod koniec filmu krótki quiz sprawdzający, jednak ta funkcjonalność jest dostępna jedynie *online*. Uczestnik szkolenia może także



Rys. 5. Przykładowy kurs i fragment filmu na portalu Coursera

rozwiązywać standardowe quizy, złożone z pytań wielokrotnego wyboru, typu „prawda – fałsz” lub „podaj odpowiedź”, oraz przysyłać pliki z rozwiązaniami zadań na platformę. Ciekawym pomysłem wykorzystanym na kursach jest tzw. *peer assessment* – wzajemne ocenianie prac przez kursantów. Po upływie terminu nadsyłania rozwiązania każdy uczestnik otrzymuje do sprawdzenia wylosowane prace pięciu innych uczestników szkolenia, a po ocenieniu prac może przystąpić do sprawdzenia własnej pracy, zgodnie z podanym przez wykładowcę szablonem punktacji. Coursera oferuje także rozbudowane fora dyskusyjne z podforami definiowanymi przez wykładowcę oraz systemem flagowania postów (wnoszące coś do dyskusji, nieodpowiadające tematyce forum).

Na jesieni 2013 roku platforma Coursera oferowała kilka szkoleń dotyczących nauki programowania: **Introduction to Systematic Program Design** – programowanie w języku funkcyjnym DrRacket, **An Introduction to Interactive Programming in Python** – programowanie prostych gier w języku Python, **Algorithms, part II** – zaawansowane szkolenie dotyczące algorytmiki, realizowane w języku Java.

Obie platformy, Udacity i Coursera, oferują również system certyfikacji. Zwykle dostępne są dwa typy certyfikatów: ukończenie szkolenia oraz ukończenie szkolenia z wyróżnieniem. Gotowy certyfikat w postaci pliku pdf jest udostępniany po pewnym czasie od ukończenia szkolenia.

A co z uczniami?

Zanim rozpoczniemy pracę z uczniami, warto zastanowić się, jakiego języka programowania chcemy uczyć. Od naszego wyboru zależy przy-

gotowanie pracowni komputerowej oraz dostępność materiałów i środowiska programistycznego dla uczniów. W szkole podstawowej i gimnazjum możemy korzystać z wizualnych języków programowania, takich jak Scratch, zaś z uczniami starszymi programować np. w języku Python. Scratch w wersji 2.0 dostępny od maja 2013 roku działa w przeglądarce internetowej i nie wymaga instalowania na komputerze. Projekty tworzone w tym języku są publikowane na stronie <http://scratch.mit.edu>, pełniącej równocześnie rolę portalu społecznościowego dla wszystkich użytkowników Scratcha. Środowisko i język są dostępne w języku polskim, na stronie projektu umieszczono kilka wideosamouczków.

Istnieją wersje instalacyjne języka Python dla wszystkich najczęściej używanych systemów operacyjnych (Windows, Mac OS X, Linux), jednak jeśli nie chcemy z uczniami programować grafiki, możemy korzystać z kompilatorów *online* np. <http://ideone.com> lub <http://codepad.org>. Na stronie <http://www.pythontutor.com> znajduje się doskonałe narzędzie pozwalające w sposób graficzny interpretować wykonywanie algorytmów zapisanych w języku Python.

Przygotowując się do lekcji, możemy wykorzystać gotowe materiały dostępne w Internecie, jednak nie zawsze są one dostosowane poziomem do wiedzy uczniów. Często również zakres przekazywanej przez nie wiedzy oraz prezentowane przykłady nie odpowiadają naszym oczekiwaniom. Jedynym rozwiązaniem jest wówczas przygotowanie własnego kursu, opublikowanie go i udostępnienie uczniom. Niezwykle istotne jest, by korzystający z takich materiałów uczeń nie czuł się pozostawiony sam sobie. Jego praca i aktywność podczas nauki zdalnej powinna być nadzorowana przez nauczyciela, dlatego opublikowanie materiałów do

```

1 # from: http://www.ece.uci.edu/~chou/py02/python.ht
2 def InsertionSort(A):
3     for j in range(1, len(A)):
4         key = A[j]
5         i = j - 1
6         while (i >= 0) and (A[i] > key):
7             A[i+1] = A[i]
8             i = i - 1
9             A[i+1] = key
10
11 input = [8, 3, 9, 15, 29, 7, 10]
12 InsertionSort(input)
13 print(input)

```

Frames

- Global variables: InsertionSort, input
- Function: InsertionSort(A)
- Object: list [8, 3, 9, 15, 29, 7, 10]
- Object: InsertionSort (local)
 - A: [8, 3, 9, 15, 29, 7, 10]
 - j: 1
 - key: 3
 - i: 0

Rys. 6. Wizualizacja algorytmu sortowania przez wstawianie

czytania i słuchania w Internecie jest zwykle niewystarczające. Powinniśmy umieścić je na platformie umożliwiającej przesyłanie rozwiązań, sprawdzanie wiedzy za pomocą testów oraz wymianę informacji np. na forum. Wygodnym rozwiązaniem może być skorzystanie z platformy e-learningowej Moodle. Jest ona dostępna bezpłatnie, stosunkowo prosta w obsłudze i ma duże możliwości.

Zanim zacniemy przygotowywać własne kursy dla uczniów, możemy przyjrzeć się rozwiązaniom innych. Wiele szkół posiada własną platformę Moodle, zawierającą kursy tematyczne poszerzające wiedzę, lub po prostu używa jej do kontaktów z uczniami i jako miejsce do udostępniania dodatkowych materiałów na lekcje.

Szkolenia *online* dla młodzieży są oferowane także przez różne instytucje oświatowe.

Ośrodek Edukacji Informatycznej i Zastosowań Komputerów w Warszawie od kilku lat oferuje specjalne szkolenia *online* dla uczniów zainteresowanych programowaniem w języku Logo. Szkolenia te mają ułatwić uczniom szkół podstawowych i gimnazjów przygotowanie do konkursów informatycznych LOGIA i miniLOGIA organizowanych na Mazowszu, zdobycie niezbędnej wiedzy matematycznej i rozwijanie zainteresowań. W roku szkolnym 2012/2013 dostępne były dwa szkolenia: **Programowanie w Logo dla uczniów szkół podstawowych** oraz **Programowa-**

nie w Logo dla uczniów gimnazjum. Rozpoczęły się one krótkim szkoleniem wstępnym (Żółw liczy i rysuje) dla wszystkich chętnych uczniów. Osoby, które uzyskały minimum 70% punktów z zadań, były zapisywane automatycznie na szkolenie **Programowanie w Logo**. Szkolenie wstępne przeprowadzone w całości metodą zdalną oferowało uczniom wiele aktywności (fora, zadania, quizy), z których w dużej części korzystali pod nadzorem nauczyciela. Szkolenie właściwe składało się z dwóch spotkań stacjonarnych (na początku oraz na zakończeniu) i trzech bloków na platformie, obejmujących około 6 tygodni zajęć. Materiały były odsłaniane stopniowo.

W każdym bloku uczniowie otrzymywali dostęp do dokumentów tekstowych z krótkimi wyjaśnieniami dotyczącymi omawianych zagadnień, przykładowych aplikacji lub materiałów animowanych, słowników pojęć, zadań do wykonania oraz quizów. Bloki były podsumowywane przez zadania programistyczne i dodatkowe.

Od roku 2013 OEIiZK oferuje także szkolenia dla uczniów, dotyczące programowania w języku Python.

Kurs na Moodle – źródło sukcesu

Zastanówmy się, jak powinien wyglądać kurs umieszczony na platformie e-learningowej. Jak



Rys. 7. Platforma eKlasa Ośrodka Edukacji Informatycznej i Zastosowań Komputerów w Warszawie

dobrac materiały? Jak zaplanować strukturę? Jakie elementy są niezbędne, by szkolenie zakończyło się sukcesem?

Oto najważniejsze zasady, którymi powinniśmy się kierować:

- Dobrze przemyśleć strukturę szkolenia, właściwie dobrać kolejność realizowanego materiału.
- Podzielić materiał na niezbyt rozbudowane bloki. Każdy blok powinien zawierać opisy omawianych zagadnień, ćwiczenia do samodzielnej pracy, dodatkowe wskazówki, zadania do wykonania. W miarę możliwości urozmaicać formę zaliczeń – pliki do przesyłania, quizy, fora, tworzone przez uczestników słowniczki itp.
- Publikować dużo przykładów, w tym przykładowe rozwiązania ćwiczeń wykonywanych samodzielnie przez uczniów. Warto dodawać do rozwiązań wyjaśnienia i wskazówki.
- Publikowane dokumenty nie mogą zawierać błędów. Materiały powinny mieć podobną formę edycyjną i najlepiej, by były przygotowane na potrzeby konkretnego szkolenia.
- Unikać „przeladowania” bloków treścią, wrzucania wszystkich posiadanych materiałów. Jeśli w bloku jest zbyt dużo treści, uczeń może mieć problem z wyszukaniem najważniejszych informacji.
- Unikać publikowania „ciężkich” plików. Jeśli na potrzeby szkolenia opracowaliśmy film wideo lub nagraliśmy wypowiedź, to warto zadbać o możliwość pobrania tych plików na komputer użytkownika zamiast odtwarzania ich na stronie. Dobrze, by pliki te miały jak najmniejszą objętość. Wygodnie też, by osoby, które wolą czytać niż słuchać, miały do dyspozycji dokument tekstowy zawierający wyciąg informacji przekazywanych w formie graficznej i dźwiękowej.
- Zastanowić się nad wyznaczaniem czasu na przesłanie rozwiązań. Dobrze jest pozwolić poprawiać wielokrotnie zadania. Czasem warto także umożliwić oddawanie rozwiązań po terminie.
- Wyjaśniać wątpliwości, a przede wszystkim komentować prace. Zawsze trzeba napisać, co jest zrobione źle – można też dodawać wskazówki, jak to poprawić. Nawet do prac, które uzyskały pełną punktację, warto dodać komentarz – może być to pochwała lub wskazówka, jak można udoskonalić rozwiązanie.
- Zachęcać do kontaktów, aktywnego udziału w szkoleniu i wymieniania informacji (niestety rzadko udaje się zmotywować grupę do wypowiedzi na forum i zadawania pytań).

Nie możemy zapominać, że szkolenia zdalne nie są wcale takie proste, jak wydaje się o tym świadczyć powszechna opinia. Przygotowanie kursu na platformie wymaga o wiele więcej pracy, niż przygotowanie się do lekcji. Przede wszystkim to, co normalnie możemy dopowiedzieć widząc, że dane zagadnienie sprawia uczniom problem, na kursie zdalnym nie tylko powinniśmy szczegółowo i jasno opisać, ale na dodatek musimy przewidzieć możliwe problemy uczniów. Na lekcjach zwykle nie czytamy szczegółowo rozwiązań wszystkich zadań wykonanych przez każdego ucznia. Na pewno niektóre zagadnienia jest łatwiej pokazać niż opisać.

Sytuacja słuchacza kursu również wcale nie jest taka łatwa. Po pierwsze, to co uważane jest zwykle za podstawową zaletę szkoleń online – uczyć się w dowolnym czasie i w dowolnym miejscu – często okazuje się ich wadą. Na lekcje przychodzimy regularnie. Jesteśmy w pracowni komputerowej w szkole, w której zazwyczaj nic nas nie rozprasza i nie przeszkadza skupić się na nauce. Tymczasem podczas szkolenia zdalnego musimy sami wygospodarować czas na naukę. W każdej chwili ktoś może nam przeszkodzić, żądając poświęcenia mu uwagi. Po drugie, od uczestnika jest wymagane samodzielne wykonanie o wiele większej liczby ćwiczeń niż na zwykłych lekcjach. Po trzecie, brak jest osobistego kontaktu z innymi uczestnikami. Po czwarte, zadania rozwiązujemy samodzielnie i jeśli mamy jakiś problem, to nie stoi koło nas nauczyciel, który szybko coś podpowie i wskaże błąd. Po piąte, w najważniejszej chwili oczywiście znów może nam nie działać Internet i nie zdążymy wykonać na czas zadanej pracy. I tak dalej, i tak dalej...

Problemy związane z udziałem w kursach online i prowadzeniem takich zajęć można mnożyć. A kursy dotyczące programowania są na pewno trudniejsze niż inne szkolenia organizowane w tej formie. Jednak mają one jedną przewagę nad różnymi kursami i samouczkami dostępnymi w Internecie – NAS, prowadzących szkolenie. To my pomagamy w doborze materiału, wyjaśniamy wątpliwości i udzielamy wskazówek. Nie zostawiamy ucznia sam na sam z trudnościami, pomagamy znaleźć błędy, chwalimy, gdy uda mu się rozwiązać problem. Kursy programowania nie są proste, ale jeśli wystarczy nam zapału, to dają wiele satysfakcji zarówno uczestnikom, jak i prowadzącym zajęcia.

Agnieszka Borowiecka jest nauczycielem konsultantem w Ośrodku Edukacji Informatycznej i Zastosowań Komputerów w Warszawie.