

Kodowanie w chmurze

Grażyna Chmielewska

Marzy mi się szkoła w chmurze

Od wielu lat pracuję w chmurze (*cloud computing*) i doceniam jej zalety: nieograniczony dostęp do zasobów w dowolnym miejscu i czasie, korzystanie z wielu darmowych aplikacji czy dzielenie się wiedzą. Chmura daje nam narzędzia do komunikacji i współpracy, pozwala dzielić się pomysłami oraz wspólnie tworzyć. Nie martwię się o zakup oprogramowania, jego instalowanie i utrzymywanie na własnych komputerach czy szkolnych serwerach. Umiejętność pracy w chmurze staje jedną z podstawowych umiejętności we współczesnym świecie. I chociaż w szkole, w której pracuję, przeważają tradycyjne metody nauczania – a może przez to z większą determinacją – postanowiłam na potrzeby zajęć z informatyki, zarówno lekcyjnych, jak i pozalekcyjnych, stworzyć swoją **minichmurę**.

Zaczęliśmy od prostych zadań domowych i testów na szkolnej platformie e-learningowej Moodle, wykorzystując ją również jako forum pomocy, wymiany doświadczeń i aktualności. Do komunikacji korzystamy z poczty na własnym serwerze pocztowym (zagwarantowane bezpieczeństwo dla uczniów w wieku 9-13 lat). Moje wcześniejsze obawy: czy sobie poradzimy (wiek uczniów, klasy integracyjne) szybko się rozwiązały – uczniowie pracowali sprawnie. Oczywiście były różne problemy, największe z samodzielną rejestracją uczestników kursów czy z logowaniem, ale to zawsze okazuje się trudne. Dzieciom najbardziej

podobały się testy *online*: określony czas na rozwiązanie, wielokrotna liczba podejść i natychmiastowa ocena. Zaczęły doceniać możliwość dostępu do prac wykonanych w domu czy szkole oraz pomocy dydaktycznych opracowanych przeze mnie.

Prowadzenie platformy e-learningowej, przygotowywanie materiałów szkoleniowych, testów, kryteriów oceniania jest niezmiernie czasochłonne. Osoby, które pracują na platformie Moodle, chyba się ze mną zgodzą. W sieci jest coraz więcej różnych serwisów i platform edukacyjnych oferujących gotowe narzędzia, które możemy wykorzystać w pracy z dziećmi do nauki i realizacji pomysłów.

Obecnie z moimi uczniami korzystamy z wielu narzędzi i aplikacji webowych. Uczymy się również programować w Scratchu, Logo, od początku bierzemy aktywny udział w międzynarodowej akcji Hour of Code. Tworzymy filmy, prezentacje, komiksy, biuletyny, infografiki, gry, animacje, wirtualne książeczki, a nasze ulubione metody pracy to metoda projektów i *blended learning* (połączenie tradycyjnych metod nauczania z aktywnościami prowadzonymi zdalnie przy pomocy komputera).

Dominują projekty interdyscyplinarne, których realizacja pochłania wiele godzin, np.: nagradzany w wielu konkursach projekt „Cyfrowe Opowieści”, „Spotkania ze Scratchem” oraz ostatni realizowany projekt „I TY możesz być programistą”. W zależności od tematyki zadania korzystamy ze szkolnej



Rysunek 1. Pasjonujące zajęcia z komputerem

platformy Moodle lub ogólnie dostępnych platform edukacyjnych, np. scratch.mit.edu lub code.org. Niektóre zagadnienia przygotowuję w formie odwróconej lekcji, korzystając z platform TED-ed lub Blendspace.

Realizacja każdego projektu stwarza uczniom okazję do odniesienia sukcesu i zaprezentowania swoich osiągnięć, każdemu projektowi towarzyszą bowiem konkursy, pokazy i prezentacje prac przed publicznością, mogą to być również spotkania z rodzicami, podczas których **dzieci uczą rodziców** programowania („Spotkania ze Scratchem”), czy **festiwale filmowe online** („Cyfrowe Opowieści”).

Nasze działania są promowane poprzez strony realizowanych projektów. Prace uczniów publikowane są również na stronie szkoły www.sp73.waw.pl, mojej stronie poświęconej informatyce informatyka.sp73.waw.pl oraz w różnych serwisach np.: YouTube, Vimeo, Scribd, Issuu, Prezi, Google+.

W ten sposób pokazuję uczniom i rodzicom, że technologie TIK służą nie tylko rozrywce i komunikowaniu się, lecz są doskonałym narzędziem do poznawania świata, ludzi, zjawisk, rozwiązywania w twórczy sposób zadań i problemów z różnych dziedzin. Pomagają również w odkrywaniu ukrytych talentów, rozwijaniu zainteresowań, by z czasem stały się pasją. Stwarzają nam praktycznie nieograniczone możliwości.

Staram się być dla moich uczniów przewodnikiem i doradcą, który wskazuje drogę rozwoju, pomaga w zdobywaniu kompetencji potrzebnych na dalszych etapach kształcenia i w przyszłym życiu zawodowym. Jedną z takich kluczowych umiejętności jest **umiejętność programowania**.

Każdy powinien się uczyć programowania!

Nauka pisania programów, to gimnastyka dla mózgu. Pozwala wypracować umiejętność efektywnego myślenia o rzeczach niezwiązanych z informatyką – powiedział Bill Gates i zgadzam się z nim w 100%.

O potrzebie nauki programowania już w szkole podstawowej mówi się ostatnio dużo, a będzie jeszcze więcej za sprawą MEN i zaproszenia do publicznej debaty (do 31 października 2015) nad projektem nowej podstawy programowej z informatyki, która zakłada naukę programowania już od pierwszej klasy szkoły podstawowej². Pewnie wielu nauczycieli stanie przed dylematem, jakie środowisko programistyczne wybrać dla swoich uczniów. Na szczęście nie musimy już uczyć trudnego dla wielu Pascala, a do dyspozycji młodych programistów jest wiele bardzo interesujących propozycji np.: Scratch, Logo, Blockly, Balty czy **Hour of Code** („Godzina Kodowania”), której jestem entuzjastką i którą polecam do realizacji szczególnie w ramach zajęć lekcyjnych.

Kodowanie przez cały rok

Hour of Code to największe wydarzenie w historii nauki. Podczas Tygodnia Edukacji Informatycznej dziesiątki milionów studentów z całego świata uczą się kodowania, rozwiązując łamigłówki na platformie code.org. W tym roku odbędzie się już trzecia edycja akcji Hour of Code w terminie 7-13 grudnia 2015. Kursy na platformie code.org są dostępne przez cały rok i można z nich swobodnie korzystać. Jednak żeby uczestniczyć w akcji, trzeba się zarejestrować i ćwiczyć kodowanie przynajmniej raz w ciągu tego tygodnia.

W „Godzinie Kodowania” biorę udział ze wszystkimi klasami 4-6 od I edycji (2013). Podstawowe pojęcia z programowania poznajemy, rozwiązując zadania, których bohaterami są postaci z popularnych gier – niesforna świnka Angry Birds i zjadający rośliny Zombie. Zabawa przypomina układanie puzzli (błoczek, podobnie jak w Scratchu): wybie-

² Więcej na stronie MEN <https://men.gov.pl/pl/jakosc-edukacji/edukacja-informatyczna/rada-do-spraw-informatyzacji-edukacji>

ramy odpowiednie elementy języka (puzzle), przeciągamy je na planszę i układamy w określonym porządku. W każdej chwili można podejrzeć kod swojego rozwiązania (w języku JavaScript). Zadaniom towarzyszą wideoporadniki, w których znane osobistości, np. Bill Gates, Mark Zuckerberg, Drew Houston (Dropbox), Chris Bosh (koszykarz, gwiazda NBA) tłumaczą podstawowe kwestie związane z programowaniem.

Hour of Code pokochałam od samego początku, a uczniowie są po prostu zachwyceni. Dlaczego? Ponieważ nauka programowania odbywa się w sposób wizualny i intuicyjny, nie wymaga zaawansowanej wiedzy matematycznej. Natomiast rozwiązywanie zadań wyzwała ogromne emocje, a każdy działający kod daje uczucie zwycięstwa, zadowolenia i wiary we własne możliwości intelektualne. Zajęcia lekcyjne przeciągają się na całą przerwę, nawet obiad nie jest w stanie oderwać uczniów od komputera. Niektórzy przyjeżdżają do szkoły godzinę, dwie wcześniej, żeby przed zajęciami rozwiązywać kolejne zadania kursu. Inni oferują swoją pomoc osobom mającym trudności, pełnią rolę moich asystentów. Bywa niekiedy, że czwartoklasista naprowadza na właściwy tok myślenia koleżanki i kolegów z klas szóstych. Kodujemy wszyscy na lekcjach i na przerwach, i w każdej wolnej chwili. Kodowaniu towarzyszy współpraca i zdrowa rywalizacja, okrzyki radości, a nawet brawa w przypadku wyjątkowo trudnych zagadek. Pokonując kolejne etapy kursu, uczniowie zdobywają trofea – brązowe, srebrne i złote puchary, oczywiście wirtualne, łącznie 27. Ukończenie całego kursu jest równoważne z uzyskaniem certyfikatu Hour of Code z podpisem Hadi Partovi (inicjator Hour of Code), który można opublikować na swoim profilu na Facebooku, Twitterze czy szkolnej stronie WWW lub wydrukować i uroczysto wręczyć na forum szkoły.

Warto podkreślić, że od II edycji Hour of Code (2014) oferta kursów jest bardzo bogata. Mamy możliwość wyboru materiałów ćwiczeniowych odpowiednio dostosowanych do wieku i możliwości uczestników, a w zależności od czasu, jaki chcemy przeznaczyć na naukę kodowania i miejsca: szkoła czy praca własna w domu, możemy wybierać między przyspieszonym kursem progra-

mowania, kursami 20-godzinnymi z podstaw informatyki dla określonych grup wiekowych i kursami tematycznymi dla wszystkich. Uczyć się może każdy, nawet od 4 lat, jeden z kursów jest bowiem specjalnie przygotowany dla dzieci, które jeszcze nie potrafią czytać. Bardzo mi się przydał, dzięki niemu mógł uczyć się kodowania jeden z moich uczniów z poważnymi dysfunkcjami. Nie widziałam chyba bardziej szczęśliwego dziecka na lekcjach informatyki.

Kolejnym argumentem przekonującym do korzystania z platformy code.org jest to, że jest ona nie tylko zbiorem materiałów ćwiczeniowych dla uczniów, ale również pomocy metodycznych, takich jak poradniki, scenariusze lekcji, a nawet gotowych pomocy dydaktycznych dla nauczycieli, opracowanych przez amerykańskich ekspertów. W razie potrzeby nauczyciele mogą korzystać z konsultacji, forum czy bezpośredniej rozmowy z fachowcami. Platforma pozwala także na tworzenie wirtualnych klas i monitorowanie osiągnięć uczniów przez nauczyciela.

Jej uniwersalność, możliwość organizowania zajęć prowadzonych różnymi metodami i w dogodnym czasie oraz atrakcyjność i możliwość indywidualizowania zadań dla uczniów zadecydowały o wyborze tej właśnie platformy do realizacji mojego projektu „**I Ty możesz być programistą**”, który jest propozycją zajęć lekcyjnych z przedmiotu zajęcia komputerowe dla klas 4-6 (9-13 lat), prowadzonych w formie lekcji oraz metodą e-learningową w ramach pracy domowej.

Kodowanie nie tylko w chmurze – informacje o projekcie i jego realizacji

Celem projektu jest:

1. Zmotywować i zachęcić uczniów do nauki programowania.
2. Przekonać, że nauka programowania nie jest trudna i nudna, może być świetną zabawą i atrakcyjną ofertą zagospodarowania wolnego czasu, rozwijającą umie-

jętności i zainteresowania oraz ciekawą alternatywą nawet dla pasjonujących gier komputerowych.

3. Uświadomić, jak ważną i potrzebną, zarówno na dalszych etapach kształcenia, jak i w przyszłym życiu zawodowym, jest w XXI wieku umiejętność programowania.
4. Zainteresować i przybliżyć uczniom zawód informatyka programisty.

Projekt obejmuje trzy cykle zajęć: pierwszy cykl zajęć wspólny dla wszystkich klas, mający na celu wprowadzenie podstawowych pojęć z programowania oraz dwa cykle zajęć oddzielnych dla poszczególnych poziomów klas, czyli zestawy zadań z Godziny Kodowania.

Na etapie opracowywania projektu miałam wiele wątpliwości, jedną z poważniejszych był wybór odpowiedniego kursu dla poszczególnych poziomów klas. Klasy 5 i 6 miały już za sobą początki kodowania, musiałam więc zdecydować się na kurs, w którym byłyby nowe zadania. Do wyboru miałam dwa: Course 2 z dość prostymi zadaniami i Course 3 (trudniejszy, polecany dla starszych dzieci), przy czym w obu kursach instrukcje i poradniki filmowe były w języku angielskim. I to właśnie budziło moje największe obawy. Ponieważ przede wszystkim zależało mi, żeby zachęcić uczniów do nauki programowania i żeby nauka nie była stresująca, zdecydowałam się na zestaw łatwiejszych zadań dla wszystkich, natomiast kurs trudniejszy przeznaczyłam dla chętnych na szóstkę. W trakcie realizacji okazało się, że wybór był trafny. Zestaw na szóstkę rozwiązało około 10% uczniów klas 5-6. Jeszcze większy dylemat miałam w przypadku materiałów dla klas czwartych. Ostatecznie wybrałam 20-Hour Intro Course, gdyż jako jedyny w całości był przetłumaczony na język polski. Nie znałam jeszcze dobrze uczniów klas czwartych, poza tym jest to grupa dzieci, które rozpoczęły naukę jako sześciolatki i obawiałam się, że bariera językowa może na długo zniechęcić młodych uczniów do nauki programowania. Teraz bogatsza o doświadczenia wiem, że nie był to dobry wybór. Kurs dla większości dzieci okazał się za trudny, tylko nieliczni rozwią-



Rysunek 2. Ciężka praca zakończona sukcesem

zali wszystkie zadania samodzielnie. Ma to i dobre strony, w ten sposób odkryłam kilka ukrytych talentów informatycznych.

Kolejnym, może nawet poważniejszym problemem do rozwiązania okazało się to jak wprowadzić podstawowe pojęcia z programowania np. algorytm, pętla, funkcja, czy potrzebne do rozwiązywania zadań nieznanne jeszcze uczniom własności figur geometrycznych. Najlepszym sposobem byłyby ćwiczenia praktyczne. No właśnie, ale jak to zrobić na zajęciach komputerowych? Zdecydowałam się na bardzo odważne rozwiązanie – cykl zajęć bez komputera, w czasie których uczniowie wykonując konkretne działania w świecie rzeczywistym wykorzystywali też różne aktywności fizyczne (chodzenie, obracanie, podskoki) i tworzyli bazę dla nowych umiejętności wykorzystywanych w świecie wirtualnym – modelowanie pojęć. I tu kolejny dylemat: jak zareagują uczniowie? Czy nie będą protestować – lekcje informatyki bez komputerów?

Okazało się, że były to dla nich najatrakcyjniejsze zajęcia. Po zakończeniu projektu wręcz się dopominali o dalsze zajęcia w formie „aktywności bez komputera” (tak je nazwałam).

Aktywność bez komputera to cykl zajęć składający się z pięciu tematów, które realizowałam w ramach 5 godzin lekcyjnych w klasach 4-6 i jednej lekcji *online* w formie tzw. odwróconej lekcji.

1. Nowe i ważne słowo: ALGORYTM – lekcja 1
2. Czy warto uczyć się programowania? – lekcja *online*
3. Zaczynamy kodowanie – lekcja 2
4. Nowe i ważne słowo: PĘTLA – lekcja 3 i 5
5. Studio graficzne – rysujemy i mierzymy kąty – lekcja 4

Pierwsza lekcja miała na celu zapoznanie uczniów z pojęciem algorytmu, uświadomienie powszechności stosowania algorytmów oraz rozwijanie umiejętności przedstawiania algorytmów w postaci opisów słownych oraz pisemnej, za pomocą schematu i kodu.

Już sam temat zajęć „Nowe i ważne słowo: Algorytm” był dla uczniów intrygujący i zachęcał do pracy. Może uczniowie znają i potrafią zdefiniować pojęcie algorytmu – warto im na to pozwolić. U mnie w odpowiedzi na zadane pytanie zapadła cisza, tylko dwóch uczniów z klas 6 próbowało wyjaśnić, co kryje się za tym słowem, kojarząc je jednak wyłącznie z matematyką. W takiej sytuacji podałam gotową definicję – najprostszą zrozumiętą dla uczniów: „Algorytm to lista czynności potrzebnych do wykonania zadania” z wyraźnym podkreśleniem, że nie musi to być wcale zadanie matematyczne, mogą to być zadania, które wykonujemy w codziennym życiu, np. przechodzenie przez ulicę. Wspólnie ustalaliśmy kolejne czynności, jakie musimy wykonać, chcąc przejść przez ulicę, po czym sprawdziliśmy, czy nasz opis słowny jest prawidłowy, porównując z zapisem algorytmu na slajdzie prezentacji, którą przygotowałam do lekcji. W podobny sposób omówiliśmy algorytm „parzenie herbaty” i „pranie brudnego ubrania w pralce”. Uczniowie podawali swoje przykłady, szukaliśmy, czy w szkole nauczyciele i uczniowie korzystają z jakichś algorytmów, np. chcąc porównać wyniki nauczania poszczególnych uczniów, całych klas, a po „odkryciu” średniej ocen sprawnie nam poszło budowanie schematu blokowego.

Wszystkie ćwiczenia opisane powyżej były rozgrzewką przed zadaniem głównym – budowaniem algorytmu za pomocą przyjętych znaków, będącego instrukcją dla malarza, jak krok po kroku ma się poruszać po planszy, żeby uzyskać efekt przedstawiony na rysunku. Uczniowie pracowali w parach, każdy zespół otrzymał kartkę z graficzną ilustracją treści zadania i listą znaków zastępujących określone czynności: „iść w prawo”, „iść w lewo”, „iść do góry”, „iść do dołu”, „zamaluj”. Później sprawdzaliśmy poprawność i liczbę rozwiązań, dyskutowaliśmy, które jest najlepsze, jakim jednym słowem (czasownikiem) określilibyśmy budowanie

algorytmów i czy komputer potrafiłby wykonać zadanie, gdyby rozumiał nasze kody. W wyniku dyskusji doszliśmy do następujących wniosków: Jeżeli istnieje kilka rozwiązań problemu, to do realizacji wybieramy rozwiązanie najkrótsze i zarazem poprawne. Budowanie algorytmów to kodowanie i komputer rozumie nasze polecenia, jeżeli będą one napisane w jakimś języku zrozumiałym dla komputera, czyli kodzie. W takim razie **programowanie** to pisanie algorytmów w języku zrozumiałym dla komputera. Lekcja zakończyła się ćwiczeniami sprawdzającymi zdobyte umiejętności, polegającymi na dopasowywaniu algorytmu do rysunku, utworzonymi za pomocą aplikacji Web 2.0 LearningApps.org.

To był dobry moment, żeby przedstawić uczniom zadanie domowe, które przygotowałam w formie lekcji *online*. Mając jedną godzinę zajęć tygodniowo, trudno wszystkie ważne tematy realizować w szkole. Wówczas wspieram się tzw. odwróconą lekcją, którą publikuję na TED-ed lub Blendspace. Uczniowie w domu, korzystając z materiałów wcześniej przeze mnie przygotowanych (filmy, artykuły, proste ćwiczenia) zapoznają się z tematem, dzięki temu w szkole mamy czas na dyskusje, stawianie pytań czy rozwiązywanie problemów z wykorzystaniem nowych wiadomości i umiejętności.

Temat: **Czy warto uczyć się programowania?** był idealny do przeprowadzenia lekcji *online*, do której przygotowałam:

- Cztery krótkie filmy: „Co to jest programowanie?”, „Każdy może uczyć się programowania”, „Czy warto uczyć się programowania?” (2 części).
- Artykuł „5 powodów by uczyć dzieci programowania”, opracowany przez zespół projektu Mistrzowie Kodowania³, który jest wprawdzie skierowany do rodziców, ale moi uczniowie, nawet w klasie 4 świetnie poradzili sobie ze zrozumieniem tekstu. Dlatego bardzo polecam.

³ <http://mistrzowiekodowania.pl>

- Listę ważnych pytań: Co to jest programowanie? Na czym polega praca informatyka programisty? Czy zawód programisty jest potrzebny w XXI wieku i dlaczego? Czy każdy może uczyć się programowania? Czy chciałbyś uczyć się programowania? Czy warto uczyć się programowania?

Prosiłam, żeby uczniowie w domu zorganizowali krótki seans filmowy i zaprosili rodziców, przeczytali artykuł i spróbowali odpowiedzieć na ważne pytania dotyczące programowania. Dlaczego angażowałam rodziców? Ponieważ bardzo często spotykam się z brakiem zrozumienia, jak ważna jest umiejętność programowania i celowość zdobywania tej umiejętności już w szkole podstawowej. Jeżeli mówimy o programowaniu, pierwszą reakcją ze strony rodziców jest obawa, czy to nie jest za trudne dla dzieci, a kolejną, do czego to jest potrzebne. Rodzice widząc dzieci korzystające z platform typu code.org czy scratch.mit.edu sądzą, że jest to tylko zabawa i nie zdają sobie sprawy, że mają one charakter dydaktyczny. Dlatego tak ważne jest włączanie rodziców w proces edukacji chociażby poprzez wspólne oglądanie filmów, podsuwanie ciekawych artykułów czy organizowanie wspólnych zajęć, np. rodzinnych warsztatów z programowania.

Większość uczniów bardzo przejęła się swoją misją i rzeczywiście w domach cała rodzina uczestniczyła w lekcji *online*. Na kolejnych zajęciach wszyscy zgodnym chórem twierdzili, że warto uczyć się programowania. Z uzasadnieniem było trudniej, przeważały odpowiedzi, że bardzo im się to przyda w życiu i przyszłej pracy zawodowej. Trafilo się kilka wypowiedzi „perłek”, oto jedna z nich: *programista to taki ktoś co ma super moc, może zrobić wszystko, co mu przyjdzie do głowy, oczywiście za pomocą komputera*. To mnie zupełnie usatysfakcjonowało. Rozpoczęliśmy naszą naukę kodowania. Każdy uczeń otrzymał „tajną karteczkę” z loginem, hasłem i linkiem do wirtualnej klasy. Ustaliliśmy zasady pracy, terminy kolejnych etapów kursu oraz sposób oceniania postępów uczniów. Pokazałam również dziennik nauczyciela wirtualnej klasy, uświadamiając w ten sposób, że nauczyciel ma pogląd pracy uczniów i możliwość monitorowania ich postępów. Warto także podkreślić zalety pracy *online*:

każde dziecko pracuje w odpowiednim dla siebie tempie i dogodnym czasie, zarówno w szkole, jak i w domu, czytelni czy w każdym innym miejscu, w którym jest dostęp do Internetu. Poza tym, na to należy zwrócić uwagę, zadania Hour of Code można rozwiązywać nie tylko na komputerach, ale na wszystkich urządzeniach mobilnych typu tablet, smartfon czy iPhone.

We wszystkich kursach Hour of Code jest wiele zadań, do których rozwiązania niezbędne są wiadomości dotyczące podstawowych własności figur płaskich. Dlatego nasza kolejna aktywność bez komputera poświęcona była odkrywaniu tych własności kwadratów i trójkątów: równobocznego oraz prostokątnego równoramiennego. Każdy zespół otrzymał przygotowany wcześniej zestaw figur: kwadrat, 3 trójkąty równoboczne, 3 trójkąty prostokątne równoramienne oraz czystą kartkę. Uczniowie bawili się w malarza z kursu i ręcznie rysowali kwadrat oraz trójkąt równoboczny.

I tu powstał problem: o jaki kąt trzeba się obrócić, rysując np. trójkąt równoboczny. Wówczas przeprowadziliśmy serię ćwiczeń: odrywanie, porównywanie, sumowanie kątów, podczas których uczniowie znajdują sumę kątów wewnętrznych w trójkącie, odkrywają, że trójkąt równoboczny ma wszystkie boki i kąty równe, znajdują kąt przyległy do kąta wewnętrznego w trójkącie i kwadracie. Po tych ćwiczeniach samodzielnie rozwiązywanie zadań z malarzem nie sprawiało już żadnych problemów.

Rozwiązując początkowe zadania, uczniowie zauważyli, że w kodach niektóre polecenia (bloczki) powtarzają się wiele razy, przez co zapis kodu strasznie się wydłuża. Sami zaczęli się zastanawiać, czy nie można zapisać krócej. Była to świetna okazja do wprowadzenia drugiego nowego i ważnego słowa: pętla. Zaczęliśmy od sterowania uczniem ochotnikiem, np. „idź do przodu 1 krok”, „idź do przodu 1 krok” i tak powtórzyliśmy z 5 razy. Bardzo szybko doszliśmy do wniosku, że znacznie prościej byłoby wydać polecenie „powtórz 5 razy idź do przodu”. W ten sposób odkryliśmy, że wielokrotne powtarzanie akcji to pętla „powtórz”. Prosiłam uczniów do szukania pętli w innych sytuacjach, np. w tańcu. Oglądając filmy z układami tanecznymi



Rysunek 3. Trwa praca zespołowa

(wcześniej nagrałam je z grupą uczniów) szukaliśmy powtarzających się elementów układu. Następnie zapisywaliśmy oglądane układy w formie algorytmu, postępując się symbolami graficznymi. W kolejnym zadaniu uczniowie, pracując w zespołach, samodzielnie tworzyli bardzo proste układy taneczne, zapisując je w formie algorytmów. Zapisy miały być czytelne, ponieważ trzeba było na koniec zatańczyć układ opracowany przez inny zespół. Oczywiście dzieci chciały utrudnić zadanie przeciwnikom i powstały układy nawet z pętlami zagnieżdżonymi. Chyba nie muszę opisywać, jakie emocje towarzyszyły naszym działaniom, dodam tylko, że lekcja tak się spodobała moim uczniom, że umówiliśmy się na dodatkowe zajęcia.

Podsumowanie

Moi uczniowie polubili kodowanie. Ledwie zakończyliśmy realizację projektu „I TY możesz być programistą”, a już pytali o następną edycję Hour

of Code. Bardzo mnie to cieszy, gdy widzę jak się rozwijają, pokonując kolejne stopnie trudności. Uczniowie pracujący szybciej, z bardziej rozwiniętym logicznym myśleniem sięgają po kolejne kursy dostępne na platformie code.org., ale także spontanicznie pomagają kolegom, którzy ślabiej sobie radzą z kodowaniem. Również dzieci mające kłopoty z nauką na innych lekcjach tu pracują z zaangażowaniem, stają się pewniejsze. Programowanie dodaje im skrzydeł i pozwala uwierzyć we własne możliwości.

Na zakończenie przytoczę fragment odpowiedzi jednego z uczniów na pytanie, czego się nauczył podczas kodowania: *kodowanie daje mi dużo wiedzy i choć jestem dopiero w czwartej klasie, to jestem w domu ekspertem z informatyki.*

Grażyna Chmielewska jest nauczycielem informatyki w Szkole Podstawowej z Oddziałami Integracyjnymi nr 73 im. Króla Stefana Batorego w Warszawie.