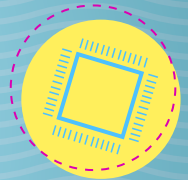
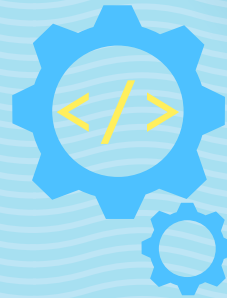
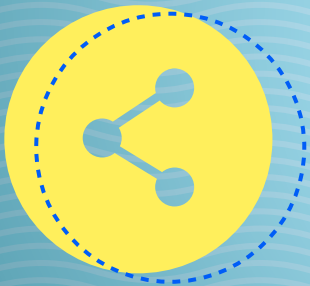


TADEUSZ NOWIK

PRZYGOTOWANIE NAUCZYCIELI DO PRAKTYCZNEGO WDRAŻANIA ELEMENTÓW PROGRAMOWANIA



MATERIAŁY SZKOLENIOWE DLA NAUCZYCIELI
SZKOŁY ĆWICZEŃ Z OBSZARU INFORMATYKI





PROGRAMOWANIE W PODSTAWIE PROGRAMOWEJ DLA SZKOŁY PODSTAWOWEJ

Istotną zmianą w podstawie programowej wdrożonej rozporządzeniem Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w zakresie ogólnie rozumianej nauki informatyki jest wprowadzenie programowania na wszystkich etapach edukacji. Nowe przepisy wprowadzają tematykę programowania już w pierwszej klasie szkoły podstawowej i kontynuują ją przez pozostałe etapy edukacji. Jest to nowa sytuacja, w której uczeń będzie przez 12 lat uczył się programowania. Jak mądrze ten czas wykorzystać i czego na danym etapie uczyć?

Podstawa programowa zakłada dla wszystkich etapów edukacyjnych takie same cele ogólne kształcenia informatycznego. Opis wymagań szczegółowych ma charakter spiralny (przyrostowy) – na każdym etapie edukacyjnym wymaga się od uczniów umiejętności zdobytych na wcześniejszych etapach i rozszerza się je o umiejętności nowe.

Cele kształcenia – wymagania ogólne¹:

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.
- III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi, w tym znajomość zasad działania urządzeń cyfrowych i sieci komputerowych oraz wykonywania obliczeń i programów.
- IV. Rozwijanie kompetencji społecznych, takich jak komunikacja i współpraca w grupie, w tym w środowiskach wirtualnych, udział w projektach zespołowych oraz zarządzanie projektami.
- V. Przestrzeganie prawa i zasad bezpieczeństwa. Respektowanie prywatności informacji i ochrony danych, praw własności intelektualnej, etykiety w komunikacji i norm współżycia społecznego, ocena zagrożeń związanych z technologią i ich uwzględnienie dla bezpieczeństwa swojego i innych.

W przypadku edukacji wczesnoszkolnej mocny akcent jest położony na stopniowe włączanie kształcenia informatycznego w sposób ściśle skorelowany z różnymi edukacjami. Ma to bezpośredni zapis w podstawie programowej:

Edukacja polonistyczna

pkt 3. pisze notatkę, życzenie, ogłoszenie, zaproszenie, podziękowanie, list; zapisuje adres nadawcy i odbiorcy; pisze krótkie teksty, wykorzystując aplikacje komputerowe.

Edukacja matematyczna

pkt 2. układa zadania i je rozwiązuje, tworzy łamigłówki matematyczne, wykorzystuje w tym procesie własną aktywność artystyczną, techniczną, konstrukcyjną; wybrane działania realizuje za pomocą prostych aplikacji komputerowych.

pkt 4. odczytuje godziny na zegarze ze wskazówkami oraz elektronicznym (wyświetlającym cyfry w systemie 24-godzinnym); wykonuje proste obliczenia dotyczące czasu; posługuje się jednostkami czasu: doba, godzina, minuta, sekunda; posługuje się stoperem, aplikacjami telefonu, tabletu, komputera; zapisuje daty, np. swojego urodzenia lub datę bieżącą; posługuje się kalendarzem; odczytuje oraz zapisuje znaki rzymskie co najmniej do XII

Edukacja przyrodnicza

pkt 3. czyta proste plany, wskazuje kierunki główne na mapie, odczytuje podstawowe znaki kartograficzne map, z których korzysta; za pomocą komputera, wpisując poprawnie adres, wyznacza np. trasę przejazdu rowerem;

Edukacja plastyczna

pkt 5. powiela za pomocą kalki, tuszu, farby, stempla wykonanego np. z korka i innych tworzyw, a także przy pomocy prostych programów komputerowych;

pkt 9. tworzy przy użyciu prostej aplikacji komputerowej np. plakaty, ulotki i inne wytwory.

Ważnym elementem uczenia informatyki jest rozwijanie tzw. myślenia komputacyjnego, czyli upraszczając uczenie rozwiązywania problemów z różnych dziedzin ze świadomym wykorzystaniem metod i narzędzi wywodzących się z informatyki.

Na myślenie komputacyjne składają się następujące umiejętności:¹

- Formułowanie problemów – rozpoznawanie, nazywanie problemów, zadawanie odpowiednich pytań.
- Zbieranie danych – określanie rzetelności danych i wiarygodności źródeł informacji.
- Rozkładanie na części – porządkowanie danych, dzielenie zadań na mniejsze.
- Rozpoznawanie schematów – klasyfikowanie (tworzenie zbiorów), rozpoznawanie podobieństw, znajdowanie istotnych i nieistotnych różnic, uogólnianie.
- Abstrahowanie i tworzenie modeli – usuwanie zbędnych informacji, upraszczanie.
- Tworzenie algorytmów – ustalanie kolejnych kroków i tworzenie zasad, sekwencja, rekurencja (powtarzalność procedur i czynności).
- Wykrywanie i diagnozowanie błędów – wyszukiwanie, znajdowanie i analizowanie błędów.
- Zrozumiałe i skuteczne komunikowanie się – formułowanie zrozumiałych komunikatów, dostosowanych do odbiorcy (komputera lub innych ludzi), kodowanie, przedstawianie (symbole i znaki).
- Ocenianie – rozpoznawanie kryteriów wartościowania, określanie priorytetów, ocenianie prototypów i rozwiązań.
- Logiczne myślenie – wyciąganie wniosków, rozpoznawanie błędów logicznych, argumentowanie.



PROGRAMOWANIE BEZ KOMPUTERA

W edukacji wczesnoszkolnej wskazane jest wprowadzenie uczniów w nowe zagadnienia informatyczne przy wykorzystaniu przestrzeni klasy, organizując aktywność dzieci z wykorzystaniem liczmanów, gier planszowych, materiału naturalnego czy form plastycznych, technicznych wykonanych przez dzieci¹. Taki zapis w podstawie programowej jednoznacznie sugeruje rozpoczęcie wdrażania programowania bez użycia komputera, przy pomocy gier, mat edukacyjnych i innych pomocy dydaktycznych. Programowanie przy pomocy komputera to dopiero kolejny etap rozwoju kompetencji programistycznych.

Przedstawione niżej propozycje ćwiczeń nigdy nie powinny być realizowane w oderwaniu od szerszego kontekstu lekcji. Powinny być interdyscyplinarnym elementem uzupełniającym realizowane treści różnych edukacji przedmiotowych.

Ćwiczenie 1. Określanie cech obiektów.

Krótki opis ćwiczenia:

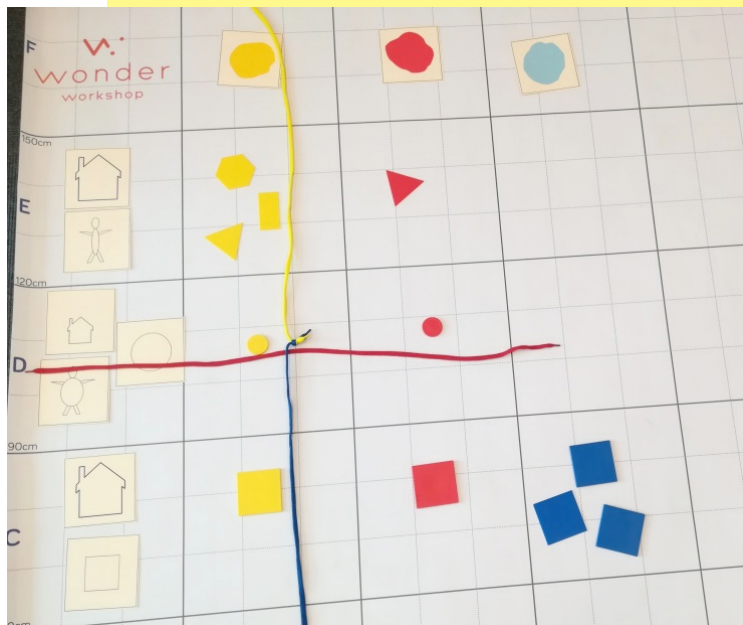
Przy pomocy maty edukacyjnej z kratownicą oraz klocków Dienesa uczniowie klasyfikują poszczególne klocki wg określonych cech i umieszczają je we właściwym polu na macie.

Cele:

- kształtowanie umiejętności klasyfikowania;
- rozwijanie umiejętności rozpoznawania schematów;
- rozwijanie zdolności dostrzegania wspólnych cech dla różnych przedmiotów;
- rozwijanie spostrzegawczości i umiejętności skupienia uwagi;
- utrwalanie pojęcia zbioru.

Sposób realizacji ćwiczenia:

Na macie w pierwszej kolumnie i w pierwszym wierszu definiujemy, jakie cechy może mieć klocek położony na przecięciu np. czerwony, cienki, duży itp. Następnie z rozsypanych klocków uczniowie wybierają kolejne klocki, opisują ich cechy i starają się wybrać pole, które spełnia te cechy. Pomocą mogą być dwie kolorowe tasiemki, które obsługuje para uczniów jak suwaki: jedna para przesuwająca tasiemkę przez kolumny, druga przez wiersze, ustawiając przecięcie we właściwym polu.



źródło: T. Nowik zasoby własne

Podsumowanie

Ćwiczenie to uczy porządkowania i klasyfikowania informacji według określonych kryteriów. Przygotowuje uczniów do wprowadzenia na kolejnych etapach edukacji zasad tworzenia klas w programowaniu obiektowym. Jest również dobrym pomysłem do zapoznania uczniów z układem współrzędnych, który pojawi się w programowaniu w Scratchu i w edukacji matematycznej. Uczy współpracy zespołowej przy obsługiwaniu tasiemek oraz tworzenia wypowiedzi w postaci opisu przedmiotu (stanu rzeczywistego, opisywania faktów) wykorzystywanego w edukacji polonistycznej.

Ćwiczenie 2. Kodowanie w ruchu.

Krótki opis ćwiczenia:

Uczniowie wspólnie wykonują sekwencyjnie czynności zakodowane pod postaciami figur geometrycznych.

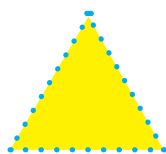
Cele:

- rozwijanie zdolności dostrzegania sekwencyjności zdarzeń;
- rozwijanie umiejętności rozpoznawania schematów;
- rozwijanie umiejętności szyfrowania i odszyfrowywania kodu;
- rozwijanie koordynacji wzrokowo-ruchowej;
- dostrzeganie regularności rytmicznych;
- utrwalenie nazw figur.

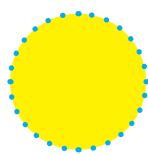
Sposób realizacji ćwiczenia:

Nauczyciel przypomina figury geometryczne: trójkąt, koło, kwadrat. Następnie wyświetla ciąg figur i razem z uczniami w ustalonym rytmie je odczytuje. Kolejnym etapem ćwiczenia jest podstawienie pod wybraną figurę czynności, którą trzeba wykonać:

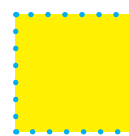
- trójkąt symbolizuje tupnięcie
- koło symbolizuje klaśnięcie
- kwadrat symbolizuje klaśnięcie w kolana



TUPNIĘCIE

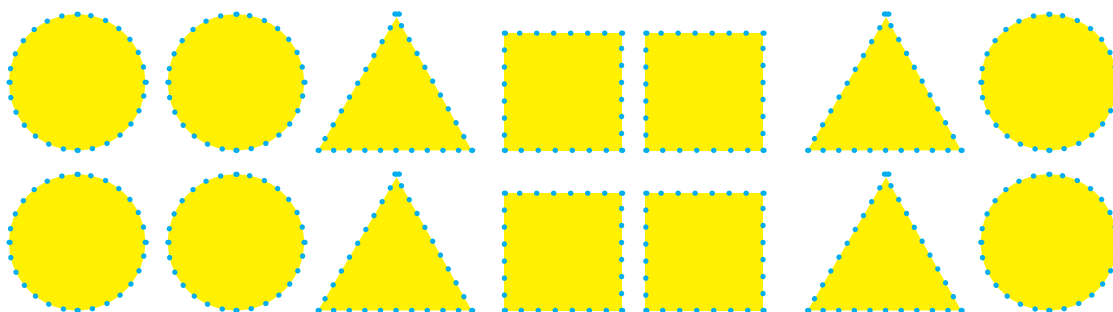


KLAŚNIĘCIE



UDERZENIE W KOLANA

Tak zakodowane czynności pojedynczo powtarzamy z uczniami. Nauczyciel wraca do wcześniej wyświetlanego ciągu figur i razem z uczniami wykonuje zakodowane czynności.



Następnie uczniowie zostają podzieleni na zespoły. W grupach sami wymyślają ciąg figur, który zapisują na dużych kartkach. Nauczyciel wiesza kartkę z kodem, a grupa odtwarza zakodowany zapis.

Powyższe ćwiczenie można zmodyfikować na różne sposoby. Poniżej dwa przykłady.

1. Rozróżnianie kierunków lewo – prawo

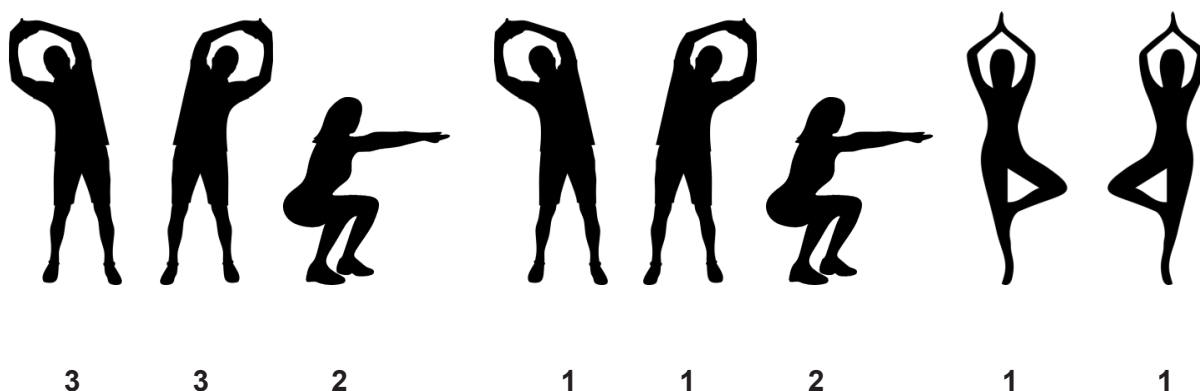
L – podnosimy lewą rękę

P – podnosimy prawą rękę

K – uderzamy w kolana

P P L L P L K K L K P K

3. Zakodowana gimnastyka. Liczby pod rysunkami mówią o liczbie powtórzeń.



źródło obrazów: <https://pixabay.com/cs/vectors> 2019.03.15

Podsumowanie

W pracy zespołowej dzieci uczą się współpracy, pełnienia określonych ról, ustalania wspólnego rozwiązania. Ćwiczenie łączy elementy z wielu dziedzin: matematyki, informatyki, muzyki, sprawności ruchowej, komunikacji. Z powtarzalnością w formie rytmu spotykamy się również w zapisie dwójkowym, który wykorzystują komputery. Zdolność wychwytywania powtarzalności i reguł jest istotną umiejętnością w programowaniu, która pozwala na upraszczanie zapisu kodu.

Ćwiczenie poza tym jest
dobrą śródlekcyjną
aktywnością ruchową.

Ćwiczenie 3. Kodujemy drogę robota Oziego.

Krótki opis ćwiczenia:

Ćwiczenie polega na sterowaniu robotem przy pomocy kodu strzałek.

Cele:

- kształtowanie umiejętności czytania i zapisywania graficznego kodu programu,
- kształtowanie umiejętności formułowania instrukcji i przetwarzanie ich w określone sekwencje,
- rozwijanie umiejętności formułowania i wykonywania prostych poleceń,
- rozwijanie orientacji przestrzennej,
- rozwijanie percepcji wzrokowej i koordynacji wzrokowo-ruchowej,
- wyszukiwanie i diagnozowanie błędów.

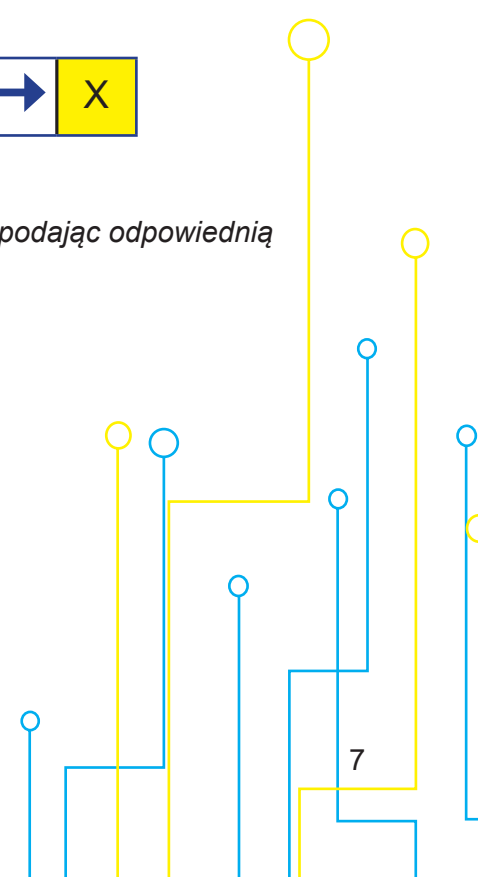
Sposób realizacji ćwiczenia:

Ćwiczenie można wykonać na macie lub na wcześniej przygotowanej karcie pracy z wrysowaną kratownicą oraz zaznaczonymi na niej elementami. Nauczyciel rozdaje uczniom karty pracy, przedstawia robota Oziego i wprowadza uczniów w jego tajną misję. Budujemy wokół ćwiczenia scenariusz, w którym osadzamy całe zadanie. Przykładem może być podanie uczniom informacji, że w niektórych państwach drony dostarczają lekarstwa i inne potrzebne środki do domów znacznie oddalonych od miast. W naszej sytuacji lekarstwa ma zanieść robot Ozi, który posiada informację o trasie zapisaną przy pomocy kodu. Zadaniem uczniów jest wrysowanie (w oparciu o przedstawiony kod) trasy robota. W zadaniu zastosowano uproszczony zapis, pomijając symbole skrętu w prawo i w lewo. Podstawowe symbole to start, strzałka i koniec. Wszystkie stosowane symbole są znakami umownymi, a więc jasno i jednoznacznie musimy określić, co one oznaczają.

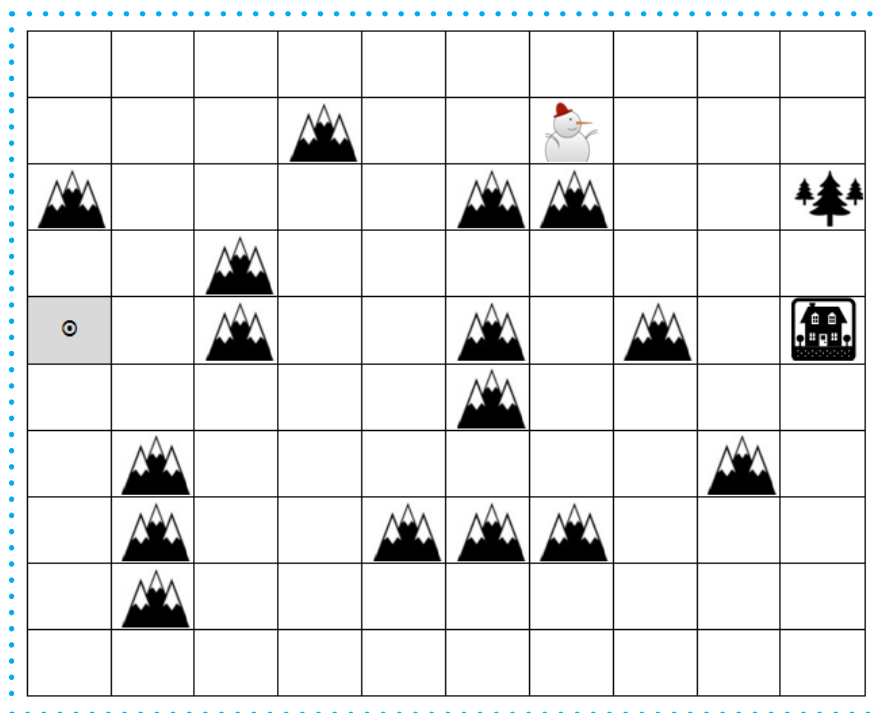
Polecenie dla ucznia: Dokąd dojdzie Ozi, jeśli będzie wykonywał dokładnie zapisany poniżej kod? Na karcie pracy wrysuj trasę, zacznij od punktu ●.



Polecenie dla ucznia: Zapisz powyższy kod w uproszczony sposób, podając odpowiednią liczbę strzałek np. 5 → 4 ↑




Jeśli mamy matę, robotem jest jeden z uczniów, a wybrany zespół steruje nim, wydając polecenia zgodnie z powyższym zapisem, np. 1 krok do przodu, 1 krok w prawo itd.



Rozszerzając to zadanie o elementy edukacji matematycznej, możemy uczniom polecić, by obliczyli, jaką drogę pokonał robot. Wówczas umawiamy się, że jedna strzałka to np. 2 km.

Polecenie dla uczniów: *Przyjmując, że strzałka oznacza 2 km, oblicz ile kilometrów pokonał Ozi.*

Na koniec robimy odwrotne zadanie polegające na napisaniu przez ucznia własnego kodu, który doprowadzi Oziego do domu. W ćwiczeniu obowiązują te same zasady. Po utworzeniu kodu uczniowie porównują go z kodem sąsiadów (czy jest taki sam, czy jest inny, czy jest dłuższy, czy krótszy itp.), wyciągają wnioski, oceniają zapisy.

Polecenie dla ucznia: *Posługując się strzałkami, napisz program, który pozwoli dojść Oziemu do domu. Musisz omijać góry. Zawsze startujesz z pola oznaczonego , a zapis programu kończysz znakiem X.*

Podsumowanie

Tego typu ćwiczenia na macie lub z pomocą karty pracy stanowią bardzo dobre wprowadzenie do zajęć z programowania w Scratchu Juniorze lub podobnych środowiskach programistycznych. Ułatwi to uczniom sterowanie duszkiem na ekranie, tworzenie prostych animacji i gier. Ćwiczenie uczy stosowania jednoznacznych poleceń oraz budowania kodu programu. Zwraca uwagę, że każdy program ma początek i koniec, a w środku instrukcje. Zadanie ma charakter interdyscyplinarny, gdyż w przykładowy temat o zdrowiu i stosowaniu leków można wprowadzić programowanie robota i przy okazji wykonać proste zadania arytmetyczne. Uczy tworzenia koncepcji, przewidywania i planowania oraz wyciągania wniosków i oceniania.

Ćwiczenie 4. „Idziemy na grzyby” – zapisywanie kodu programu.

Krótki opis ćwiczenia:

Zadaniem uczniów jest zebranie przez wyznaczonego ucznia wszystkich grzybów jadalnych i zapisanie za pomocą kodu trasy poruszania się ucznia.

Cele:

- kształtowanie umiejętności czytania i zapisywania graficznego kodu programu;
- kształtowanie umiejętności formułowania instrukcji i przetwarzanie ich w określone sekwencje;
- rozwijanie umiejętności formułowania i wykonywania prostych poleceń;
- rozwijanie orientacji przestrzennej;
- rozwijanie percepcji wzrokowej i koordynacji wzrokowo-ruchowej;
- wyszukiwanie i diagnozowanie błędów;
- rozwijanie umiejętności rozpoznawania grzybów jadalnych.

Sposób realizacji ćwiczenia:

Do ćwiczenia potrzebujemy matę, zdjęcia grzybów oraz karty z symbolami do kodowania drogi. Zdjęcia grzybów na odwrocie zawierają krótki opis: nazwa grzyba, czy jest jadalny, czy trujący, cechy charakterystyczne.

Ćwiczenie wpisuje się w temat związany z jesienią i rozpoznawaniem różnych gatunków grzybów. Na macie umieszczamy obrazki z grzybami jadalnymi i trującymi oraz miejsce startu. Jednego z uczniów wybieramy na grzybiarza. Dwóch uczniów wyznaczamy do sterowania grzybiarzem. Ustalamy z całą klasą jakich komend możemy używać. Zapisujemy je na tablicy. Przy tłumaczeniu poleceń omawiamy również karty z symbolami kodowania tak, by uczniowie wiedzieli jakiemu poleceniu odpowiada jaki symbol na karcie. Pozostałych uczniów dzielimy na grupy. Ich zadaniem będzie ułożenie obok maty kodu trasy grzybiarza. Jeśli jest zbyt mało miejsca w sali, to część uczniów może wykonać zadanie na ławce z małymi kartami do kodowania trasy.

Kod układamy od dołu do góry. Specjalnym znakiem kodujemy podniesienie grzyba.

Gdy uczeń zbierze wszystkie jadalne grzyby, kończymy zapisywanie kodu, bo cel został osiągnięty. Następnie uczniowie sprawdzają poprawność kodu, np. ustawiają innego ucznia i patrząc na zapisany kod, sterują nim.

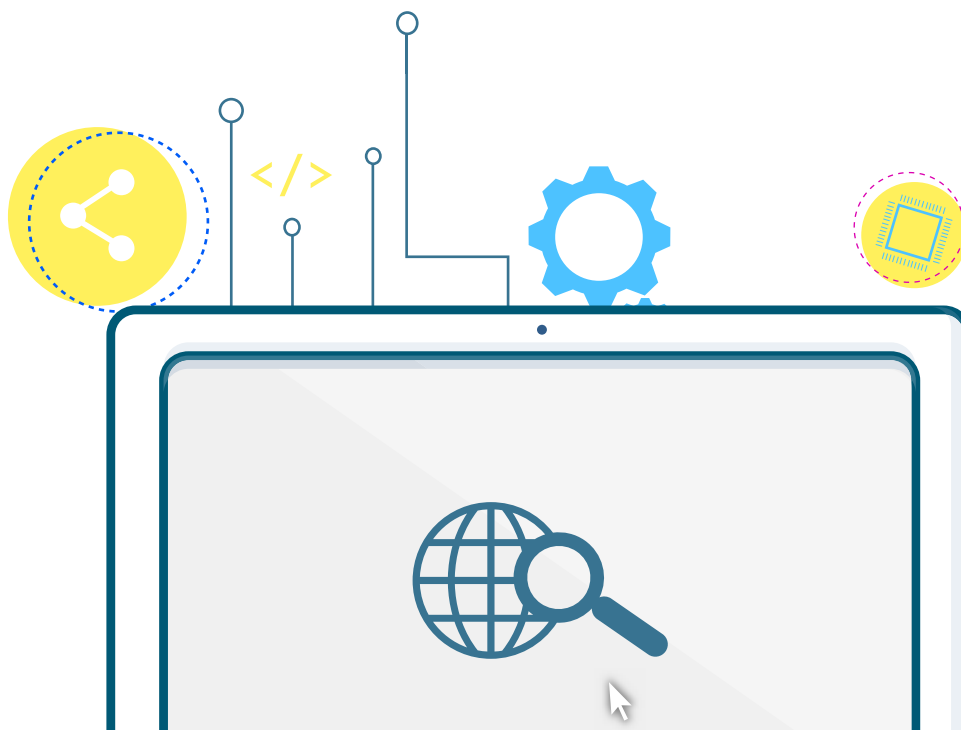
Na koniec grzybiarz rozdaje zebrane grzyby każdej grupie, która zapoznaje się z opisem umieszczonym na odwrocie. Przedstawiciele grup dzielą się odczytaną informacją z całą klasą.



źródło: T. Nowik zasoby własne

Podsumowanie

Ćwiczenie jest wprowadzeniem do kodowania na kolejnych etapach edukacji. Uczy posługiwania się instrukcjami i zapisu programu. Dodatkowo w zadaniu zastosowano znaki skrętu w prawo i w lewo. Jest to element umowny, który oznacza tylko skręt, a nie skręt i krok do przodu, co na początku może nie być tak oczywiste dla uczniów. Praca w zespole pozwala na bezpośrednie korygowanie ewentualnych błędów i wymianę zdań. Cały zapis trasy zbudowany jest z uporządkowanych jednoznacznych informacji, których realizacja prowadzi do rozwiązania problemu. W ten sposób zadanie sprzyja myśleniu algorytmicznemu i wprowadza ucznia w świat algorytmów.



Ćwiczenie 5. Wprowadzamy pojęcie pętli.

Krótki opis ćwiczenia:

W ćwiczeniu wykorzystujemy wcześniejsze doświadczenia uczniów z kodowania trasy na macie przy pomocy znaków graficznych. Ćwiczenie zwraca uwagę na powtarzalność fragmentów kodu i możliwość skrócenia zapisu przez zastosowanie pętli, w której powtarza się dany fragment kodu określoną liczbę razy.

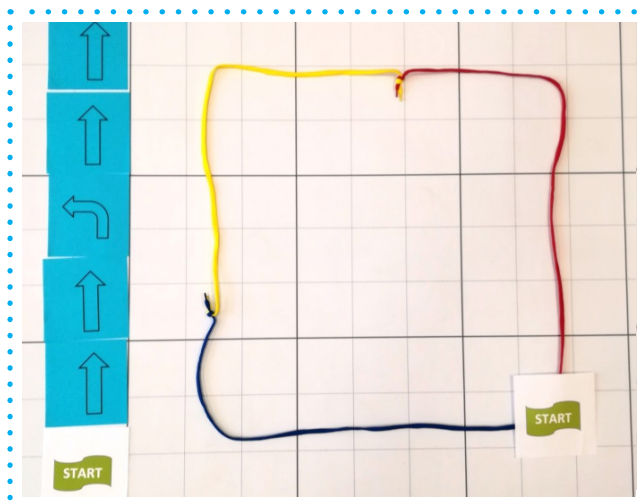
Cele:

- zrozumienie pojęcia pętli w programowaniu;
- kształtowanie umiejętności formułowania problemów;
- rozwijanie umiejętności dzielenia zadania na części i upraszczania;
- rozwijanie spostrzegawczości i dostrzegania cech wspólnych;
- kształtowanie umiejętności czytania i zapisywania graficznego kodu programu;
- kształtowanie umiejętności formułowania instrukcji i przetwarzanie ich w określone sekwencje;
- rozwijanie umiejętności formułowania i wykonywania prostych poleceń;
- rozwijanie orientacji przestrzennej;
- rozwijanie percepcji wzrokowej i koordynacji wzrokowo-ruchowej.

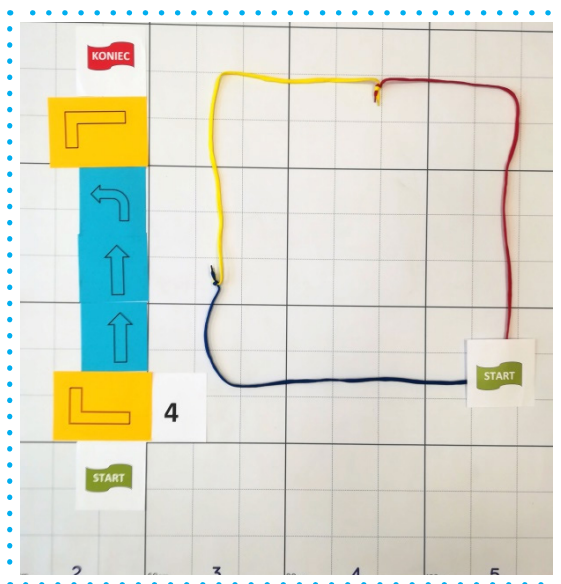
Sposób realizacji ćwiczenia:

Do wykonania ćwiczenia potrzebujemy maty, wstążki lub sznurka oraz kart z symbolami do kodowania.

Przypominamy pojęcie kwadratu, dochodząc do stwierdzenia, że wszystkie boki są równe. Następnie na macie przy pomocy wstążki wyznaczamy kontur kwadratu oraz umieszczamy punkt startu. Jeden z uczniów będzie poruszał się po wstążce zgodnie z poleceniami podawanymi przez grupę wybranych uczniów. Polecenia wcześniej są omówione z nauczycielem i zapisane, np. na tablicy. Przy tłumaczeniu poleceń omawiamy również karty z symbolami kodowania tak, by uczniowie wiedzieli, jakiemu poleceniu odpowiada jaki symbol na karcie. Kolejna grupa uczniów układa obok maty karty z symbolami do kodowania odzwierciedlające wydawane polecenia. W ten sposób zapisują drogę, jaką porusza się uczeń. Kodowanie zawsze zaczynamy od symbolu startu i kończymy symbolem końca. Pozostali uczniowie mogą wykonać kodowanie na ławce z małymi kartami do kodowania trasy. Na końcu uczniowie sprawdzają czy zapisy kodowania są poprawne.



Następnie przyglądamy się czy jakiś fragment kodu się powtarza. Wydzielamy go w kodzie, np. rozsuwając kod i dzieląc go na powtarzające się fragmenty. Liczymy ile takich powtórzeń się pojawiło. Wprowadzamy nowe karty z symbolami do kodowania, które oznaczają pętlę. Obok układamy zapis z wykorzystaniem pętli i porównujemy z zapisem bez pętli.

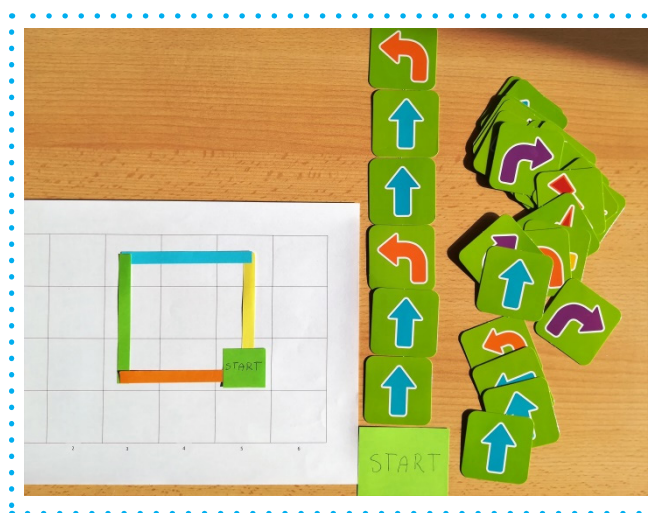


źródło: T. Nowik zasoby własne

Uczniowie sprawdzają czy zapisy wyrażają to samo. Ustawiają ucznia i próbują go przeprowadzić trasą wg zapisu z pętlą, zaznaczając, w którym momencie zapisu znajduje się uczeń odliczając powtórzenia w pętli. Uczniowie porównują dwa zapisy, wskazują różnice, oceniają, który ich zdaniem jest lepszy i dlaczego.

Na koniec możemy wyświetlić inny kod z pętlą i poprosić, by uczniowie spróbowali wg tego kodu przejść po macie.

Poniżej zdjęcie tego samego ćwiczenia wykonywanego przez uczniów na ławce z kartami pracy i małymi kartami z symbolami.



źródło: T. Nowik, kodowanie na ławce z małymi kartami

Podsumowanie

Ćwiczenie jest wprowadzeniem do zagadnień związanych ze stosowaniem pętli – jednej z trzech podstawowych konstrukcji programowania strukturalnego obok instrukcji warunkowej wyboru. Zasada dzielenia zadania na mniejsze fragmenty usprawnia pracę, pozwala zrozumieć uproszczony zapis z zastosowaniem pętli.

Ćwiczenie 6. Kodowanie z kubeczkami.

Krótki opis ćwiczenia:

Zadaniem uczniów jest ustawienie wieży z kubków według określonego schematu i zakodowanie wykonywanych czynności wg ustalonych znaków graficznych. W drugiej części ćwiczenia uczniowie budują nową wieżę z kubków, korzystając z zapisu graficznego kodu utworzonego przez innych uczniów.

Cele:

- kształtowanie umiejętności zamieniania czynności na instrukcję;
- rozwijanie umiejętności czytania graficznego kodu programu;
- rozwijanie myślenia algorytmicznego, analitycznego i logicznego;
- kształtowanie umiejętności czytania schematów i tworzenia na ich podstawie modeli przestrzennych;
- kształtowanie umiejętności analizy kodu, wykrywania i diagnozowania błędów;
- uczenie dokładności i precyzyjności w wykonywaniu zadań;
- uczenie przestrzegania ustalonych zasad;
- rozwijanie umiejętności współpracy w grupie;
- rozwijanie percepcji wzrokowej i koordynacji wzrokowo-ruchowej.

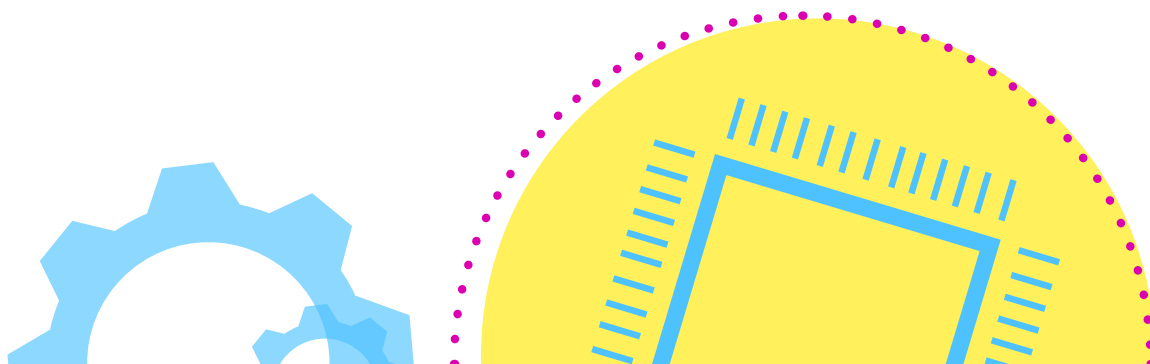
Sposób realizacji ćwiczenia:

Do ćwiczenia będą potrzebne kartonowe lub plastikowe kubeczki, gumki recepturki, sznurek, karty ze wzorami zbudowanej z kubków wieży, plansza z kołami, kartki do zapisania kodu.

Lekcję zaczynamy od zastosowania różnych poleceń i instrukcji w życiu codziennym. Uczniowie podają przykłady różnych poleceń. Uwrażliwiamy ich na dokładność formułowania poleceń i ich precyzję wykonania.

Następnie przechodzimy do zasadniczego ćwiczenia, które będzie się składało z dwóch części. W pierwszej zbudują wieżę z kubków w oparciu o jej schemat na obrazku, jednocześnie notując wykonywane czynności przy pomocy znaków graficznych. W drugiej części zbudują inną wieżę w oparciu nie o schemat graficzny, tylko zapis graficzny czynności czyli kod.

Uczniowie w grupach otrzymują kartę ze znakami graficznymi do kodowania, rysunek z wieżą ustawioną z kubków (każda grupa inny) oraz planszę z kołami, na których będą ustawiać kubki – budować wieżę. Nauczyciel objaśnia znaki graficzne – instrukcje przesuwania kubków.



Następnie ustalamy i zapisujemy na tablicy zasady, których trzeba przestrzegać, np.

- kubek zawsze ustawiamy na polu start dnem do dołu,
- kubek możemy przesuwając tylko co pół koła – to jest jeden ruch (1 strzałka),
- nie wolno przeskakiwać przez kubki, a więc zaczynamy ustawianie kubków od tych, które stoją najdalej na podkładce i na pierwszym poziomie,
- przesuwanie każdego kubka kodujemy na osobnej karcie w oddzielnych ponumerowanych wierszach, przykład:

kubek 1. →→→→→→→→↵

kubek 2. →→→→→

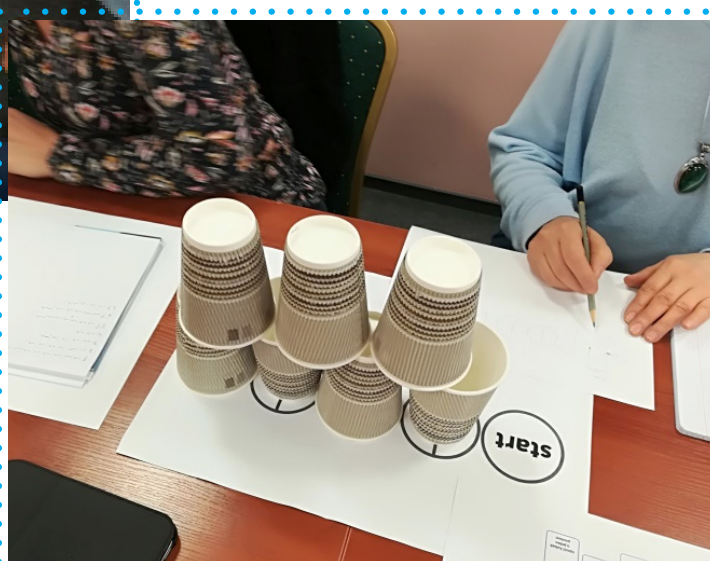
– możemy zastosować zapisy skrócone np. : 6 →↵

Zanim uczniowie przystąpią do realizacji zadania w grupach możemy na jednym przykładzie przećwiczyć je wspólnie, by wyeliminować ewentualne błędy.

Dla ćwiczenia umiejętności współpracy można posłużyć się gumkami i sznurkiem do ustawiania kubków, jak na zdjęciu poniżej.



źródło: T. Nowik zasoby własne



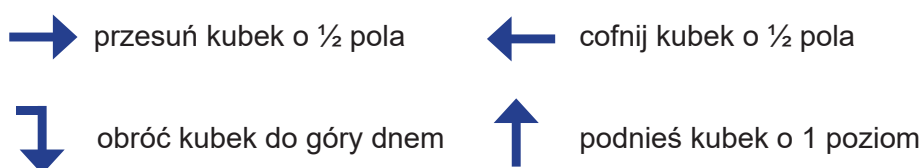
źródło: T. Nowik zasoby własne

Należy zwrócić uwagę, by uczniowie starannie zapisywali kod do każdego przesuwanego kubka, znaki były czytelne i jednoznaczne, a kod do każdego kubka w osobnym wierszu.

W drugiej części ćwiczenia uczniowie wymieniają swoje karty z kodem między grupami i na podstawie zapisu próbują odtworzyć wieżę z kubków. W ten sposób sprawdzą poprawność zapisu wykonanego przez inną grupę. W razie wykrycia błędu zgłaszają nauczycielowi i proponują prawidłowe rozwiązanie. Na koniec między grupami, które wymieniły się zapisami kodów odbywa się porównanie ze schematami wieży na kartkach i przedyskutowanie zapisów.

Dla ćwiczenia umiejętności współpracy można posłużyć się gumkami i sznurkiem do ustawiania kubków, jak na zdjęciu poniżej.

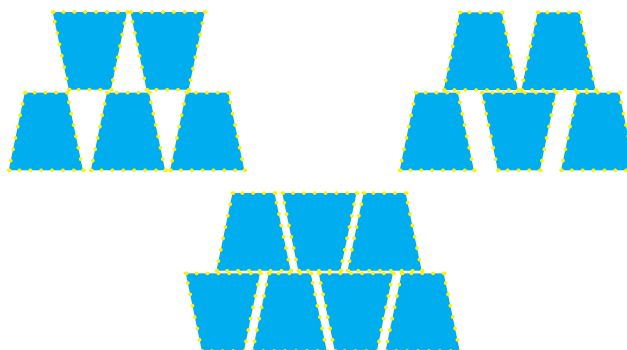
Propozycja symboli do kodowania:



Plansza



Przykładowe wzory wież do ułożenia:



Podsumowanie

Przez całe ćwiczenie uczniowie pracują w grupach, podejmując wspólne działania na rzecz rozwiązania zadania. Uczą się współpracy, wymieniają rozwiązaniami, sprawdzają zapisy i wychwytyją błędy. W informatyce wiele złożonych problemów rozwiązuje się w zespołach. To istotna umiejętność, która pozwala często na osiągnięcie stawianych celów. W ćwiczeniu istotna jest dokładność i staranność zapisu kodu, która w późniejszych etapach jest nieodzowna przy programowaniu w środowisku tekstowym. Oprócz czysto informatycznych korzyści, dzieci poprawiają koordynację wzrokowo-ruchową podczas przestawiania kubków przy pomocy gumek.



ZASOBY INTERNETOWE DOTYCZĄCE PROGRAMOWANIA

Strony z materiałami, ćwiczeniami i scenariuszami lekcji:

- <http://www.oswajamyprogramowanie.edu.pl/>
- <http://kodowanienadywanie.blogspot.com/>
- <http://maliprogramisci.blogspot.com/>
- <http://mistrzowiekodowania.pl>
- <http://wiki.mistrzowiekodowania.pl>
- <http://godzinakodowania.pl>
- <http://zabawyzprogramowaniem.edu.pl>
- www.bobr.edu.pl
- <http://programowanie-w-ruchu.pl/>

Kursy programowania dla najmłodszych

- <https://code.org/>
- <http://partners.disney.com/hour-of-code/>

Strony i platformy do programowania:

Scratch – scratch.mit.edu

Jedno z najbardziej popularnych środowisk do programowania wizualnego. W Scratchu, przy pomocy bloczków, można zaprogramować własne interaktywne historie, gry i animacje oraz podzielić się swoimi produktami online.

Blockly Games – blockly-games.appspot.com

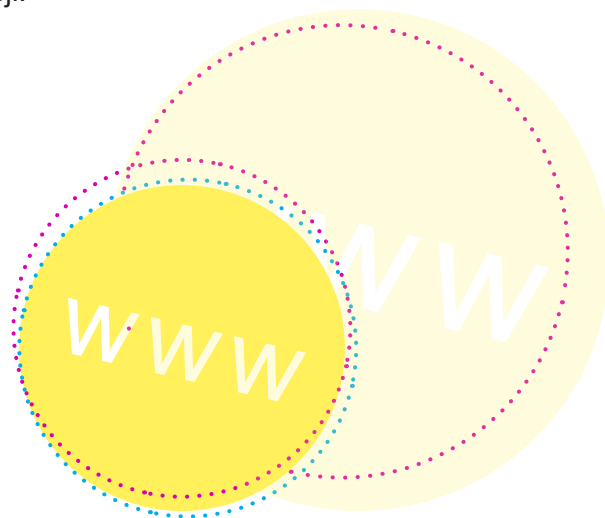
Blockly Games to seria gier edukacyjnych, które uczą programowania. Jest przeznaczona dla uczniów, którzy nie mieli wcześniej doświadczenia w programowaniu komputerowym. Z każdym poziomem trudność zadań szybko rośnie.

CodeMonkey – playcodemonkey.com

CodeMonkey to zabawne i edukacyjne środowisko gry, w którym dzieci uczą się kodować w prawdziwym języku programowania, bez wcześniejszego doświadczenia. Darmowy jest tylko okres próbny.

LightBot – lightbot.com

LightBot został zaprojektowany z myślą o początkujących programistach. LightBot to gra logiczna oparta na kodowaniu, potajemnie uczy logiki programowania podczas gry. Aplikacja na urządzenia mobilne.



Kodable – kodable.com

Kodable oferuje kilka poziomów programowania, począwszy od gier na poziomie przedszkolnym, aż po zadania uwzględniające czytanie i pisanie w języku JavaScript. Uczeń zaczyna przygodę z programowaniem od prostych poleceń związanych z nadawaniem kierunku ruchu, aż po tworzenie warunków oraz powtórzeń na wyższych poziomach trudności.

Code Combat – codecombat.com

Code Combat uczy kodowania w języku Python i JavaScript. Dzieci mogą bawić się na różnych poziomach fabularnej gry. Witryna wymaga od gracza, aby potrafił czytać i pisać, choć niektóre gotowe fragmenty kodu można wybrać z rozwijanego menu.

Gry:

- **Gra złapmy lwa** – www.zlapmylwa.pl
- **Math Duel** – aplikacja na urządzenia mobilne
- **Wieża Hanoi**

Przykładowe roboty edukacyjne:

Dash i Dot – wonderpolska.pl

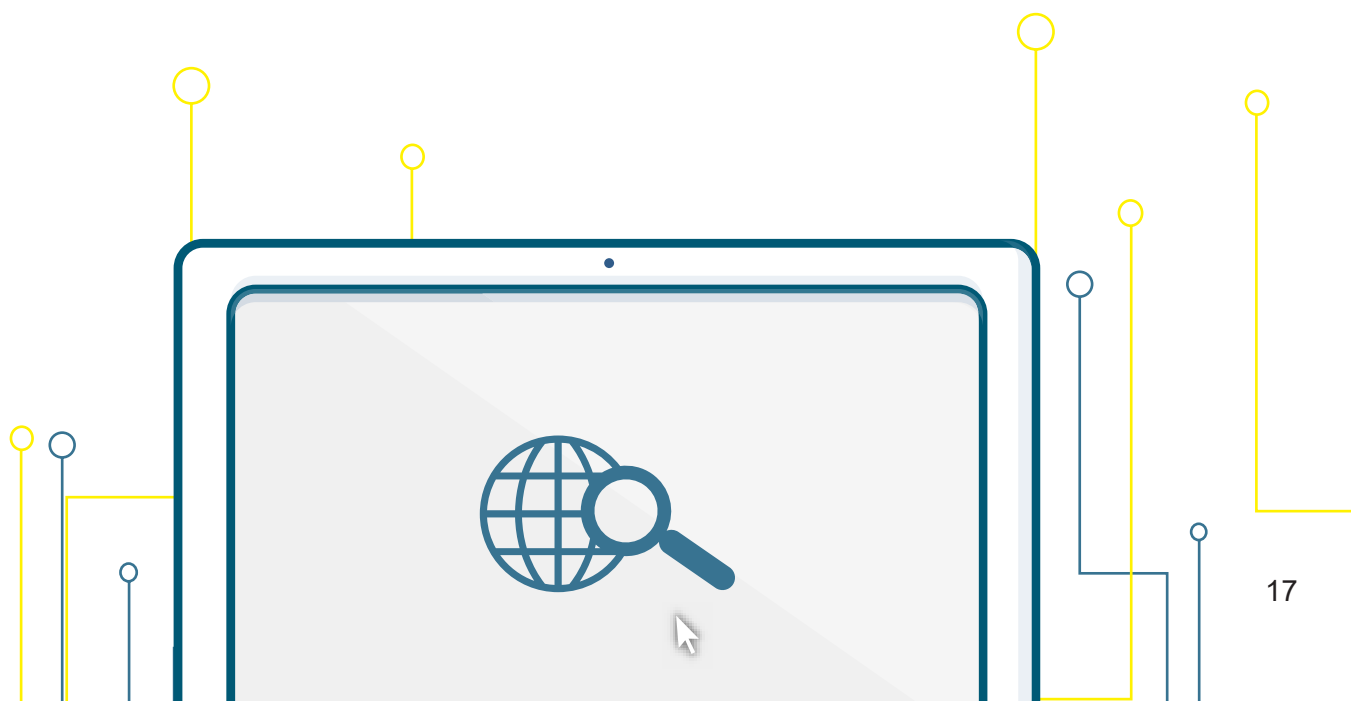
Dash i Dot to roboty edukacyjne, które wprowadzą ucznia w świat programowania poprzez zabawę. Przy użyciu urządzeń mobilnych dzieci programują Dasha, aby ten zaśpiewał, zatańczył i podróżował po całej klasie. Roboty mają sensory, dzięki którym reagują na otoczenie.

Ozobot – ozobot.pl

Mały, inteligentny robot do nauki programowania w każdym wieku. Umożliwia proste kodowanie przy pomocy kolorowych linii rysowanych na kartce papieru, kolorowych kodów, a także środowiska bloczkowego i tekstowego.

mBot – trobot.pl

mBot to solidny robot, który można dowolnie łączyć z klockami Lego. Do programowania przeznaczone jest środowisko niemal identyczne wizualnie do Scracha. Uczniowie znający Scracha bez problemu odnajdą się w programowaniu mBota.





CZĘŚĆ 4: WNIKI POJĘĆ⁴

Algorytm – sposób rozwiązania określonego problemu, zadania w skończonej liczbie instrukcji, przy użyciu odpowiednich reguł i metod podających postępowanie krok po kroku; prawidłowo zbudowany algorytm zapewnia uzyskanie za każdym razem oczekiwanego wyniku (...).

Program komputerowy – zestaw instrukcji i poleceń rozumianych przez system operacyjny i procesor, który został stworzony w celu realizacji określonych zadań.

Instrukcja – polecenie, rozkaz w językach programowania; najmniejsza część tzw. programu (kodu) źródłowego.

Procedura – część programu komputerowego odpowiedzialna za wykonanie określonego zadania.

Funkcja – część programu komputerowego, którego wywołanie (...) powoduje pewne zdarzenie.

Zmienna – zarezerwowany obszar w pamięci, służący do przechowywania wartości liczbowych lub tekstowych.

Podstawowe konstrukcje programowania⁵

Pętla – umożliwia cykliczne wykonywanie ciągu instrukcji określoną liczbę razy, do momentu zajścia pewnych warunków, dla każdego elementu kolekcji lub w nieskończoność.

Instrukcja warunkowa – jest elementem języka programowania, który pozwala na wykonanie różnych instrukcji, w zależności od tego czy zdefiniowane przez programistę wyrażenie logiczne jest prawdziwe czy fałszywe. Możliwość warunkowego decydowania o tym, jaki krok zostanie wykonany jest jedną z podstawowych własności współczesnych komputerów.



Bibliografia

¹ Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej.

² http://www.ceo.org.pl/sites/default/files/news-files/elementy_myslenia_komputacyjnego_wedlug_ceo.pdf 2019.03.15

³ Rozporządzeniem Ministra Edukacji Narodowej z dnia 14 lutego 2017 r.

⁴ M. Sławik, J. Syjud, „Szkolny słownik komputerowy”, 2002.

⁵ <https://pl.wikipedia.org> 2019.03.15

Redakcja i korekta:
Anna Busza

Opracowanie graficzne i skład:
Klaudia Karpeta

Poznań 2019

Fundacja Familijny Poznań
ul. Staszica 15
60-526 Poznań
www.familijny.pl

Materiały szkoleniowe powstały w ramach projektu nr POWR.02.10.00-00-5009/18
„Wielkopolska szkoła ćwiczeń w Cogito” w ramach Programu Operacyjnego Wiedza Edukacja
Rozwój 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Społecznego.