Informatyka

Standardy do opracowania autorskich programów zajęć pozalekcyjnych dla uczniów uzdolnionych

Szkoła podstawowa

Opracowała grupa ekspercka w składzie:
mgr Tadeusz Bury
dr inż. Jacek Dąbrowski
dr Hanna Furmańczyk
dr inż. Zbigniew Ledóchowski
mgr Ryszard Szubartowski

Koordynator grupy eksperckiej:
dr hab. Joanna Jędrzejowicz, prof. UG

Recenzent:
mgr Jarosław Drzeżdżon

Gdańsk 2011

Innowacyjny projekt systemowy *Pomorskie – dobry kurs na edukację. Wspieranie uczniów o szczególnych predyspozycjach w zakresie matematyki, fizyki i informatyki* jest współfinansowany ze środków Europejskiego Funduszu Społecznego i budżetu państwa w ramach Priorytetu IX Programu Operacyjnego Kapitał Ludzki 2007 - 2013.

DEPARTAMENT EDUKACJI I SPORTU
Urząd Marszałkowski Województwa Pomorskiego, ul. Okopowa 21/27, 80-810 Gdańsk
tel.: 58 32 68 850, faks: 58 32 68 854, e-mail: des@pomorskie.eu, www.pomorskie.eu, www.urzad

1. Wprowadzenie

Na polskim „rynku" edukacyjnym istnieją dwa środowiska programistyczne proponowane przez wydawnictwa do nauki programowania: Logo z jego aktualną wersją wspierającą wielozadaniowość – Imagine, którą na potrzeby polskiej edukacji nazwano Logomocją oraz Baltie, którego twórcą jest Bohumir Soukup. Autorzy podręczników dla szkół podstawowych i gimnazjów również proponują bądź mające dłuższe tradycje w Polsce środowisko żółwia – Logomocja, bądź jeszcze bardziej proste środowisko graficzne „klocków" wraz z duszkiem – Baltie. Ośrodki propagujące te środowiska w celu popularyzacji proponują szereg konkursów ogólnopolskich.

Logomocja

MiniLogia - dla szkół podstawowych http://minilogia.oeiizk.waw.pl

PolLogia - dla gimnazjalistów http://logo.oeiizk.waw.pl

W ostatnim czasie na stronie Logomocji pojawiły się inne konkursy o zasięgu ogólnopolskim jak: Konkurs na pokaz lub grę przygotowaną w środowisku Logomocji-Imagine, *Ja i Internet - moje bezpieczeństwo w sieci, Żaba w Logo, Tajemnice komputera, Przygody sieciaków, Czyste środowisko wokół mojej szkoły.*

Baltie

http://www.sgpsys.com/pl/

i serwer konkursowy Baltie:

http://contests.sgpsys.com/(S(itllsb45nlv1qe55jtwsqira))/Default.aspx

Swoistego rodzaju niedogodnością powyższych środowisk ograniczającą powszechny do nich dostęp jest koszt zakupu licencji dla szkół, nauczycieli, uczniów.

Biorąc pod uwagę tę niedogodność proponujemy wprowadzenie do zajęć środowiska programistycznego SCRATCH zaprojektowanego przez Mitchela Resnicka (między innymi pomysłodawcę serii zabawek Lego MindStorms i twórcę języka StarLogo). Środowisko to jest wspierane przez MIT (Massachusetts Institute of Technology), a jego istotną zaletą, poza ową rekomendacją MIT, jest istnienie polskiej wersji językowej oraz nieodpłatność licencji. Nie bez znaczenia jest też polskojęzyczne wsparcie na internetowej stronie – „Jak rozpocząć poznawanie Scratch-a i strony sieciowej Scratch-a?": http://info.scratch.mit.edu/pl/Support

Co istotne, zajęcia z programowania w szkołach podstawowych z wykorzystaniem praktycznie wszystkich powyższych środowisk (ze szczególnym uwzględnieniem SCRATCH'a) można prowadzić wg poniższego układu ogólnie sformułowanych haseł tematycznych:

- ruch (zmiana położenia „duszka”),
- animacje (tworzenia ruchomych obiektów),
- rysowanie,
- dźwięki,
- powtórzenia poleceń (pętle),
- obiekty,
- zmienne i wyrażenia,
- wprowadzanie danych,
- instrukcje warunkowe.

Programy zajęć pozalekcyjnych z informatyki dla szkoły podstawowej winny przede wszystkim uwzględniać specyfikę tego etapu edukacyjnego. Realizowane w tej chwili w szkole podstawowej zajęcia programowe (a w jeszcze większym stopniu dotyczy to zajęć realizowanych w oparciu o założenia wynikłe z wdrażanej systematycznie nowej podstawy programowej) zawierają głównie zagadnienia związane z podstawami oraz wybranymi obszarami zastosowań technologii informacyjno-komunikacyjnych. Nawet dotychczasowa nazwa przedmiotu Informatyka (w nowej podstawie programowej są to już wyłącznie Zajęcia komputerowe – za to realizowane w klasach I-VI) była w tym kontekście dość myląca. W szczególności dopuszczone do użytku szkolnego dla II etapu edukacyjnego (tym bardziej dla etapu I, dla którego zajęcia z komputerem pojawiły dopiero od roku szkolnego 2009/2010) programy zajęć informatycznych zasadniczo nie przewidywały i nie przewidują obecności treści związanych z algorytmami i programowaniem. Pojedyncze odniesienia do tematyki algorytmów mają w szkole podstawowej charakter ciekawych eksperymentów pobudzających, czy raczej wprowadzających elementy myślenia algorytmicznego. Konkursy o tematyce informatycznej organizowane dla uczniów z II etapu edukacyjnego też generalnie nie dotyczą zagadnień algorytmicznych, ale raczej wiążą się z możliwością (niekiedy bardziej zaawansowanego) wykorzystania technologii informacyjno-komunikacyjnych w różnych dziedzinach. Nieliczne wyjątki (np. konkursy MiniLogia, Balti) raczej potwierdzają regułę.

Czy zatem standardy zajęć pozalekcyjnych z informatyki, adresowanych do uczniów szkół podstawowych szczególnie tą tematyką zainteresowanych winny uwzględniać raczej kontynuację (w sposób jeszcze bardziej zaawansowany?) tematyki zajęć lekcyjnych, czy też winny proponować w postaci listy zagadnień programowych nowe treści w tym związane z problematyką wprowadzającą młodych ludzi w to co jest jednak istotą informatyki (a nie technologii informacyjno-komunikacyjnej i jej zastosowań!) czyli w obszar algorytmów i programowania? Nie przekreślając roli zagadnień związanych z zastosowaniami technologii informacyjno-komunikacyjnych proponujemy głównie tematykę zajęć odnoszącą się do obszaru tematycznego związanego ze wstępem do algorytmów i elementami programowania, ale realizowaną przy pomocy zupełnie innych (niż w

gimnazjum czy szkole ponadgimnazjalnej), wymienionych we wprowadzeniu, bardzo oryginalnych środowisk. Wydaje się, że te środowiska same w sobie mogą istotnie przyczynić się do sukcesu dydaktycznego takich zajęć, które to zajęcia, co szczególnie i ponownie warto podkreślić adresowane będą do osób taką tematyką zainteresowanych. Dzięki temu można będzie wyłonić grupę uczniów o potencjalnie większych możliwościach i predyspozycjach.

Zaleca się, aby programy takie realizowane były przede wszystkim dla uczniów z II etapu edukacyjnego (klasy IV-VI), choć nie wyklucza się po starannej analizie różnych aspektów dopuszczanie do zajęć w pojedynczych przypadkach np. uczniów klas trzecich (niekoniecznie do wszystkich zajęć, ale elastycznie stosownie do ich tematyki). Zakłada się, że z racji specyfiki podejścia (realizacja zagadnień programowych przede wszystkim w oparciu o 1-2 wybrane środowiska) zajęcia tworzyć będą w pewnej części zamknięte bloki powiązanych ze sobą logicznie treści (np. tematy wprowadzające do danego środowiska pracy), które jednak niekoniecznie muszą być zrealizowane do samego końca (zasada stopniowania trudności wymusi umieszczenie ew. treści do pominięcia w części końcowej tych bloków). Przy opracowywaniu programów należy zadbać o taką realizację tematów, aby zachowano ową logiczną kolejność, swobodny dobór kolejności pozostawia się przy tematach rekapitulujących pewną grupę treści (zadania problemowe związane ze stosowaniem poznanych technik programowania). Pamiętać przy tym należy, aby z jednej strony dobór czasu na realizację kolejnych treści był elastyczny, a z drugiej, aby w miarę możliwości stosować indywidualizację, także w tempie pracy grupy uczniów objętych zajęciami pozalekcyjnymi. Między innymi dlatego nie sugerujemy w tym opracowaniu żadnej liczby godzin przewidzianej na realizację proponowanych dalej zagadnień szczegółowych. Przyjmujemy, że z racji licznych uwarunkowań psychologiczno-metodycznych w odróżnieniu od III i IV etapu edukacyjnego nie jest to w przypadku szkoły podstawowej sprawa pierwszorzędnej wagi, ale należy ją pozostawić w dużym stopniu inwencji i doświadczeniu prowadzących zajęcia nauczycieli.

Dodatkowo, zakłada się, że osoby/instytucje przygotowujące programy zajęć dodatkowych opracują również odpowiednie elektroniczne materiały dydaktyczne, do umieszczenia na portalu edukacyjnym.

Zajęcia powinny być zaprojektowane do przeprowadzenia w grupach więcej niż 4-5 osobowych.

2. Cele ogólne zajęć

Zasadniczym celem proponowanych zajęć jest dążenie do stymulowania aktywności poznawczej i twórczej uczniów oraz stworzenie warunków do poznawania często nowych dla nich obszarów informatyki. Zajęcia powinny też zapewnić uczniom warunki do nabywania, a raczej rozwijania umiejętności poszukiwania, porządkowania i wykorzystywania informacji z różnych

źródeł. Stosowny dobór treści, metod pracy (a także proponowanych środowisk programowania) winien rozbudzać zainteresowania, ale również właściwie ukierunkować ciekawość uczniów i docelowo zachęcać (w kolejnych etapach edukacyjnych) do podejmowania jeszcze trudniejszych wyzwań. Uczeń winien uzyskiwać satysfakcję poprzez świadomość rozwiązywania problemów trudnych (dobieranych wszakże stosownie do jego możliwości). Nauczyciel powinien umożliwić uczniowi wykorzystanie jego predyspozycji do rozwijania kompetencji w zakresie samodzielnego pogłębiania wiedzy informatycznej i samodzielnego podołania wyzwaniom. Może to się wiązać z szansą weryfikacji postępów w konkursach zewnętrznych, choć z racji specyfiki tego etapu (także niewielkiej liczby stosownych konkursów) to nie musi być celem najistotniejszym. Ważniejsza jest pielęgnowanie zainteresowań uczniów i przygotowanie do ich świadomego rozwijania w przyszłości.

Ponadto, do celów ogólnych zaliczyć można również:

- rozwiązywanie problemów i podejmowanie decyzji z wykorzystaniem komputera,
- dobór poznawanych technik rozwiązywania problemów do stawianych zadań,
- kształcenie myślenia twórczego, oryginalnego i krytycznego,
- umiejętność formułowania twórczych i dociekliwych pytań, także ocen w tym dotyczących swoich rozwiązań i pomysłów.

3. Cele szczegółowe zajęć

Wiązać się będą z kolejnymi tematami. Na pewno będzie to jednak poznanie często nowych dla uczniów środowisk pracy (oprogramowania) i wybranych konstrukcji programistycznych (instrukcje warunkowe, pętle, rekurencja itp.) oraz poznanie (w stosownym zakresie) filozofii programowania (np. pojęcie obiektu) z uwzględnieniem możliwości wykorzystania nabywanej wiedzy i umiejętności w rozwiązywaniu praktycznych problemów przypisanych kolejnym zagadnieniom.

4. Zasady nauczania

W toku pracy powinny być uwzględnione potrzeby i oczekiwania uczniów, a w większym na pewno niż przy kolejnych etapach edukacyjnych stopniu jego nastawienie emocjonalne, a także predyspozycje intelektualne, a nawet fizyczne. W szczególności praca podczas zajęć pozalekcyjnych winna przebiegać zgodnie ze znanymi z dydaktyki zasadami:

1) przystępności - przekazywane wiadomości muszą być zrozumiałe dla ucznia i dostosowane do jego możliwości poznawczych
2) poglądowości - przekaz nauczyciela musi być poparty przykładami, ćwiczeniami (metody wykładowe w odniesieniu do uczniów szkół podstawowych winny przyjmować raczej formę pogadanki realizowanej w niedużym wymiarze czasowym w niezbędnych momentach)
3) stopniowanie trudności - świadome przejście od zadań prostych do coraz bardziej skomplikowanych (wyrażane także w kolejności realizacji zagadnień wprowadzających informacje teoretyczne)
4) stwarzania wyzwań – poruszane treści powinny wzbudzać zainteresowanie ucznia, być dla niego wyzwaniem, zachęcać do aktywności
5) związku teorii z praktyką - ukazywanie wykorzystania wiedzy teoretycznej w praktyce (naturalnie w praktyce pojmowanej z punktu widzenia ucznia szkoły podstawowej, gdzie często chodzi o prostą rekapitulacje poznanego faktu teoretycznego w skonstruowanym ćwiczeniu programistycznym).

Niezależnie od tego, aby nie zniechęcić uczniów należy stosować bardzo często indywidualizację procesu nauczania (zwłaszcza w doborze treści problemów i w zakresie tempa pracy).

5. Metody pracy

Specyfika przedmiotu oraz fakt, że odbiorcą zajęć jest uczeń bardziej zainteresowany realizowaną tematyką wymusza stosowanie przez nauczyciela metod nauczania, które będą aktywizowały uczniów. Wśród metod aktywizujących, które szczególnie sprawdzić mogą się w odniesieniu do uczniów szkól podstawowych wymienić należy następujące:

- „burza mózgów”
- metoda układanki (puzzli)
- metoda projektu
- metoda inscenizacji

Dzięki metodom aktywizującym w większym stopniu niż przy tradycyjnym podejściu do nauczania angażuje się różne zmysły ucznia, uczy kreatywnego myślenia, prowadzi do twórczych rozwiązań, ponadto uczniowie przyswajają sobie umiejętność uczenia się. W szczególności „burza mózgów” daje szansę na analizę różnych rozwiązań, spierania się na rację, możliwość publicznego argumentowania (co nie jest dla ucznia szkoły podstawowej szczególnie częsta sytuację w procesie nauczania), uczy pracy w zespole. Pracy w zespole uczy także metoda projektu, takie sytuacje

może nie będą w odniesieniu do proponowanej tematyki zajęć szczególnie częste, ale są możliwe do zaaranżowania. Zwrócić też warto uwagę na ostatnią z wymienionych metod. Przez inscenizację w odniesieniu do tych zajęć, rozumie się przede wszystkim kreowanie ciekawych sytuacji praktycznych, które uczeń musi rozwiązać (samodzielnie lub w zespole). Może się to przyczynić do rozwijania umiejętności samodzielnego myślenia, a także do większej atrakcyjności zajęć i na pewno jest przeciwieństwem często „nudnych" dla tak młodych uczniów wywodów teoretycznych. Jeśli taka inscenizacja zawiera w sobie element rywalizacji to działa dodatkowo motywująco na uczestników zajęć.

6. Rola nauczyciela w procesie dydaktycznym

Nauczyciel w czasie swoich zajęć z uczniami powinien:

- odpowiednio organizować pracę ucznia
- stymulować rozwój zainteresowań uczniów
- stosować odpowiednie dla swoich uczniów metody pracy
- być źródłem eksperckiej wiedzy w zakresie tematyki realizowanej podczas zajęć w tym w zakresie obsługi poznawanych środowisk programowania
- przygotowywać własne zestawy ćwiczeń oraz inne materiały do wykorzystania na zajęciach
- wykazywać stałe zainteresowanie pracą ucznia i motywować go w jego działaniach
- czuwać nad kondycją emocjonalno-intelektualną oraz fizyczną uczniów
- dostosowywać zadania do możliwości poznawczych uczniów
- stosować indywidualizację zajęć
- dążyć do osiągnięcia założonych celów
- zachęcać uczniów do samooceny i samodoskonalenia,
- zachęcać i rozwijać aktywność uczniów w tym poprzez stwarzanie im możliwości rywalizacji i kontroli swojej wiedzy i umiejętności w różnych konkursach

7. Standard opisu programu zajęć pozalekcyjnych w szkole podstawowej

Program zajęć pozalekcyjnych powinien zawierać:

- temat zajęć
- imię i nazwisko autora
- adresata (uczeń klas IV – VI lub opcjonalnie uczeń młodszy)
- wymagania wstępne – rozumiane jako temat z list zagadnień, które powinny poprzedzać dany temat
- tematy z list zagadnień, które będą poruszane w ramach danych zajęć
- czas trwania zajęć
- cele szczegółowe
- sposób organizacji zajęć obejmujący:
    1. opis problemu – treść zadania, która będzie podana uczniom na początku zajęć. Problemy, których rozwiązanie powinno gwarantować wzbudzenie zaciekawienia, zainteresowania ucznia, ilustruje pewne pojęcia lub zastosowanie pewnych konstrukcji programistycznych, albo nawiązuje do zajęć poprzednich. Może być związany z problemem rzeczywistym, albo tylko rekapitulować wcześniejszy pokaz lub wprowadzenie pojęć.
    2. teoretyczny opis problemu – tylko do wiadomości nauczyciela i tylko jeśli jest to problem rozbudowany, typu zastosowania, a nie prosta ilustracja wprowadzonych pojęć lub przećwiczenie nowych umiejętności.
    3. wskazówki wraz z opisem odnośnie użycia metod przyjętych przy rozwiązaniu problemu (przykładowe metody nauczania zostały wyszczególnione powyżej). Ostatecznego wyboru metody dokonuje nauczyciel, kierując się charakterem oraz możliwościami swoich uczniów.
    4. przykładowe scenariusze rozwiązań problemu – może być ich kilka, tak, by można było zastosować podejście oparte na indywidualizacji procesu nauczania.
    5. jeżeli temat wymaga wyjaśnienia uczniom pojęć/zagadnień im nieznanych program zajęć powinien przewidywać taką ewentualność i zawierać odpowiednie wskazówki dla nauczyciela. Formą pomocy dla nauczyciela oraz ucznia będą również elektroniczne kursy/materiały umieszczone na platformie e-learningowej.
    6. zestaw kilku (3-5) zadań-ćwiczeń do samodzielnego rozwiązania. Zadania te powinny uwzględniać różny poziom uczniów. Ponadto zaleca się przygotowanie zadań dotyczących poruszanego na zajęciach zagadnienia jak i jego rozszerzenia – tak, aby umożliwić uczniowi możliwość samodzielnego rozwoju we wskazanym na zajęciach kierunku.

DEPARTAMENT EDUKACJI I SPORTU
Urząd Marszałkowski Województwa Pomorskiego, ul. Okopowa 21/27, 80-810 Gdańsk
tel.: 58 32 68 850, faks: 58 32 68 854, e-mail: des@pomorskie.eu, www.pomorskie.eu, www.urzad

7. przykładowe rozwiązania zamieszczonych zadań - tylko do wiadomości nauczyciela.
8. przewidywane osiągnięcia ucznia.
9. spis literatury, obejmujący również adresy stron w Internecie, które mogą zostać wykorzystane na zajęciach.

8. Lista zagadnień na zajęcia pozalekcyjne

Tematy zajęć dodatkowych dla uczniów zainteresowanych w zakresie Informatyki powinny być dobierane indywidualnie przez nauczyciela. Poniższa lista zawiera bazę przykładowych tematów, które, po opracowaniu programów zajęć pozalekcyjnych, będą mogły być wykorzystane przez nauczycieli.

Jak już wspomniano wcześniej, liczba godzin, którą nauczyciel przeznaczy na zajęcia, powinna być indywidualnie dostosowana do możliwości uczniów. Lista zagadnień zawiera w pierwszej grupie bloki tematyczne dla środowiska SCRATCH (w tym na zasadzie kolejnej realizacyjnej zwłaszcza bloków tematycznych od 1 do 4), a w drugiej dla środowiska LOGOMOCJA. Naturalnie LOGOMOCJA i SCRATCH są tylko środowiskami, w których uczeń rozwiązuje problemy, a ich poznanie nie jest celem samym w sobie. W efekcie w obu środowiskach można rozwiązywać (z dokładnością do różnic związanych z ich specyfiką) podobne problemy. Daje to możliwość wykorzystania podczas zajęć obu środowisk (np. uzupełniająco), elastycznego operowania listą zagadnień, a nawet dokonywania pewnych analiz porównawczych związanych z rozwiązywanymi problemami. Przygotowując listę zagadnień starano się przynajmniej do pewnego stopnia wyeksponować te możliwości i kształtować listę zagadnień dla obu środowisk z jak największym podobieństwem (czasami brzmią one wręcz identycznie). Tam gdzie tak się stało zbieżne będą też cele dla kolejnych bloków tematycznych. Różnice między środowiskami podkreślają przede wszystkim niektóre szczegółowe problemy możliwe do zrealizowania tylko w jednym z nich (albo możliwe w sposób wygodniejszy w jednym z nich).

8. 1. Zagadnienia realizowane w środowisku scratch

Wprowadzenie do środowiska scratch

Programy zajęć powinny obejmować:

- zapoznanie z interfejsem użytkownika, podstawowe narzędzia i panele,
- sterowanie „duszkiem”, jego parametrami i innymi parametrami środowiska,

- proste kompozycje graficzne w trybie bezpośrednim,
- zasady tworzenia i uruchamiania elementarnych skryptów.

Cele szczegółowe:

- uczeń potrafi realizować proste ćwiczenia związane z operowaniem „duszkami" i ich parametrami,
- uczeń tworzy kompozycje graficzne w trybie bezpośrednim,
- uczeń potrafi realizować skrypty związane z tworzeniem prostych motywów i kompozycji.

Konstrukcje graficzne i animacyjne

Programy zajęć powinny obejmować:

- tworzenie rozbudowanych kompozycji graficznych,
- realizację skryptów „współpracujących" ze sobą w rozwiązaniu większego problemu,
- wykorzystanie animacji w rozwiązywanych problemach,
- podstawowe operacje związane z dźwiękiem, wykorzystanie efektów dźwięków i ogólniej multimedialnych w tworzonych kompozycjach.

Cele szczegółowe:

- uczeń wykorzystuje dostępne narzędzia do tworzenia zaawansowanych kompozycji,
- graficznych oraz wykorzystujących efekty multimedialne,
- uczeń wpływa na charakter swoich kompozycji poprzez umiejętne wykorzystanie,
- dostępnych parametrów i ustawień,
- uczeń realizuje zadania wymagające użycia kilku skryptów i ich współdziałania.

Sterowanie zdarzeniami. Zmienne. Instrukcje warunkowe i iteracyjne

Programy zajęć powinny obejmować:

- zapoznanie z możliwością sterowania zdarzeniami z użyciem zmiennych, instrukcji warunkowych i iteracyjnych,
- różne warianty instrukcji warunkowych i iteracyjnych, sterowanie zdarzeniami za pomocą „nasłuchiwania" klawiatury i myszy, a także czujników światła, dźwięku itp.

DEPARTAMENT EDUKACJI I SPORTU
Urząd Marszałkowski Województwa Pomorskiego, ul. Okopowa 21/27, 80-810 Gdańsk
tel.: 58 32 68 850, faks: 58 32 68 854, e-mail: des@pomorskie.eu, www.pomorskie.eu, www.urzad

- operowanie wyrażeniami, wyrażenia w warunkach logicznych,
- rozwiązywanie różnych problemów z wykorzystaniem konstrukcji programistycznych.

Cele szczegółowe:

- uczeń potrafi budować bardziej zaawansowane konstrukcje programistyczne i sterować zdarzeniami,
- uczeń rozumie znaczenie poszczególnych typów instrukcji i stosuje je zależnie od potrzeb,
- uczeń wykorzystuje zmienne i wyrażenia do realizacji swoich pomysłów,
- uczeń wykorzystuje techniki programowania wstępującego i zstępującego.

Wprowadzenie do programowania obiektowego

Programy zajęć powinny obejmować:

- definiowanie obiektów w Scratch,
- rozwiązywanie problemów z wykorzystaniem obiektów różnego typu.

Cele szczegółowe:

- uczeń rozumie pojęcie obiektu i jego znaczenie w rozwiązywanych problemach,
- uczeń stosuje elementy programowania obiektowego w tworzonych problemach.

Zaawansowane możliwości Scratch

Programy zajęć powinny obejmować:

- zapoznanie z możliwością tworzenia grafiki 3D,
- realizacja złożonych kompozycji multimedialnych z zaawansowanymi możliwościami sterowania, w tym z wykorzystaniem tzw. silnika,
- operowanie kompresją dźwięku i grafiki,
- publikowanie swoich projektów w sieci internetowej.

Cele szczegółowe:

- uczeń realizuje zaawansowane projekty multimedialne,
- uczeń umie nadać realizowanym projektom możliwie optymalny charakter,

- uczeń realizuje projekty o charakterze zespołowym.

Rozwiązywanie prostych problemów algorytmicznych w środowisku Scratch

Programy zajęć powinny obejmować:

- zapoznanie z możliwością realizacji elementarnych algorytmów w środowisku SCRATCH,
- rozwiązywanie przykładowych problemów takich jak:
  - proste algorytmy obliczeniowe i związane z geometrią,
  - problemy rekurencyjne,
  - fraktale.

Cele szczegółowe:

- uczeń potrafi zrealizować przykładowe algorytmy klasyczne w środowisku SCRATCH,
- uczeń rozumie znaczenie podstawowych konstrukcji algorytmicznych i uzyskuje wiedzę oraz umiejętności do rozwiązywania trudniejszych problemów w przyszłości.

8.2. Zagadnienia realizowane w środowisku logomocja

Wprowadzenie do środowiska Logomocja

Programy zajęć powinny obejmować:

- zapoznanie z interfejsem użytkownika, przybornik,
- sterowanie żółwiem, jego parametrami i innymi parametrami środowiska,
- proste kompozycje graficzne w trybie bezpośrednim,
- zasady tworzenia i uruchamiania procedur.

Cele szczegółowe:

- uczeń potrafi realizować proste ćwiczenia związane z operowaniem żółwiem i jego parametrami,
- uczeń tworzy kompozycje graficzne w trybie bezpośrednim,
- uczeń tworzy i uruchamia procedury realizujące proste kompozycje graficzne.

Konstrukcje graficzne i animacyjne oraz wykorzystujące dźwięk

Programy zajęć powinny obejmować:

- tworzenie rozbudowanych kompozycji graficznych z wykorzystaniem procedur,
- składanie procedur,
- wykorzystanie animacji w rozwiązywanych problemach,
- podstawowe operacje związane z dźwiękiem,
- wykorzystanie efektów dźwięków.

Cele szczegółowe:

- uczeń wykorzystuje dostępne narzędzia do tworzenia zaawansowanych kompozycji graficznych oraz wykorzystujących efekty dźwiękowe,
- uczeń wpływa na charakter swoich kompozycji poprzez umiejętne wykorzystanie,
- dostępnych parametrów i ustawień,
- uczeń realizuje zadania wymagające użycia kilku procedur i ich współdziałania.

Instrukcje warunkowe i iteracyjne

Programy zajęć powinny obejmować:

- zapoznanie z możliwością tworzenia kompozycji przy pomocy instrukcji warunkowych i iteracyjnych,
- operowanie wyrażeniami, wyrażenia w warunkach logicznych,
- rozwiązywanie różnych problemów z wykorzystaniem prostych konstrukcji programistycznych.

Cele szczegółowe:

- uczeń potrafi budować bardziej zaawansowane konstrukcje programistyczne i sterować zdarzeniami,
- uczeń rozumie znaczenie poszczególnych typów instrukcji i stosuje je zależnie od potrzeb.

Programowanie strukturalne i obiektowe w LOGOMOCJI

Programy zajęć powinny obejmować:

- procedury z parametrami, przekazywanie parametrów,

- wprowadzenie do programowania wstępującego i zstępującego,
- rozwiązywanie problemów z wykorzystaniem klas procedur,
- wykorzystanie obiektów,
- rozwiązywanie problemów z wykorzystaniem obiektów różnego typu.

Cele szczegółowe:

- uczeń rozumie pojęcie obiektu i jego znaczenie w rozwiązywanych problemach,
- uczeń potrafi analitycznie (syntetycznie) rozważać stawiane problemy i wskazywać plan ,
- rozwiązania problemu,
- uczeń stosuje elementy programowania obiektowego w tworzonych problemach.

Zaawansowane możliwości i zaawansowane problemy w LOGOMOCJI

Programy zajęć powinny obejmować:

- zapoznanie z możliwością rozwiązywania problemów rekurencyjnych,
- realizacja złożonych kompozycji graficznych (fraktale),
- praca w trybie tekstowym,
- struktury danych i ich wykorzystanie w rozwiązywaniu problemów.

Cele szczegółowe:

- uczeń stosuje rekurencję przy rozwiązywaniu problemów graficznych i tekstowych,
- uczeń potrafi tworzyć znane (płatek Kocha, dywany Sierpińskiego itd.) konstrukcje fraktalne oraz rozwiązywać problemy wymagające zaprojektowaniu innych konstrukcji tego typu,
- uczeń stosuje wybrane struktury danych (listy, słowa) w rozwiązywanych problemach.

Rozwiązywanie prostych problemów algorytmicznych w środowisku LOGOMOCJA

Programy zajęć powinny obejmować:

- zapoznanie z możliwością realizacji algorytmów klasycznych w środowisku LOGOMOCJA, w tym wymagających struktur danych,

- rozwiązywanie problemów z konkursów związanych z algorytmami i programowaniem w LOGOMOCJI.

Cele szczegółowe:

- uczeń potrafi zrealizować przykładowe algorytmy klasyczne w środowisku LOGOMOCJI,
- uczeń rozumie znaczenie podstawowych konstrukcji algorytmicznych i uzyskuje wiedzę oraz umiejętności do rozwiązywania trudniejszych problemów w przyszłości.

9. Literatura

Przy tworzeniu programów zajęć zaleca się korzystanie z poniższych pozycji bibliograficznych:

1. Aho, J. Hopcroft, J. Ullman, Projektowanie i analiza algorytmów, Helion 2003.
2. L. Banachowski, K. Diks, W. Rytter, Algorytmy i struktury danych, WNT 1996.
3. T. H. Cormen, C. E. Leiserson, R. L. Rivest , C. Stein, Wprowadzenie do algorytmów, WNT Warszawa 2004.
4. S. Dasgupta, Ch. Papadimitriou, U. Vazirani, Algorytmy, PWN 2010.
5. Drozdek, Wprowadzenie do kompresji danych, WNT 2007.
6. D. Harel, Y. Feldman, Rzecz o istocie informatyki. Algorytmika, WNT 2008.
7. M. Jankowski, Elementy grafiki komputerowej, WNT 2006.
8. B.W. Kernighan, D. M. Ritchie, Język ANSI C, Helion 2010.
9. M. Kutyłowski, W.B. Strothmann, Kryptografia. Teoria i praktyka zabezpieczania systemów komputerowych, RM 1999.
10. Z. Michalewicz, Jak to rozwiązać czyli nowoczesna heurystyka, WNT Warszawa 2006.
11. Ross, Ch. Wright, Matematyka dyskretna, PWN 1999.
12. Schneier, Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C, WNT 2002.
13. Sysło, N. Deo, J. Kowalik, Algorytmy optymalizacji dyskretnej, PWN 1995.
14. J.D. Ullman, J. Widom, Podstawowy wykład z systemów baz danych, WNT 1997.
15. V. Vazirani, Algorytmy aproksymacyjne, WNT 2004.
16. Wilson, Wprowadzenie do teorii grafów, PWN 2008.
17. K. Wojtuszkiewicz, Programowanie strukturalne i obiektowe, PWN 2009.