programowanie wizualne scratch 2.0

piotr szlagor

enauczanie.com

Autor: Piotr.Szlagor.net

Spis treści

Wstęp
- Programowanie w szkole
- O Scratchu
- Opinia studentów o Scratchu
- Scratch a Logo: Porównanie kryterialne

Przygotowanie do pracy
- Co nowego w Scratchu 2.0?
- Zakładanie konta w serwisie Scratch
- Rozgrzewka

Pierwsze kroki ze Scratchem
- Okno programu
- Strona projektu
- Zaczynamy programować
  - Tworzymy pierwszy program
  - Kalejdoskop
  - Obliczanie pola kwadratu
- Zmienne i listy
  - Zmienne
  - Listy
- Tworzenie własnych bloczków
- Komentarze
- Publikowanie programów na stronach internetowych

Rysowanie ze Scratchem
- Kwadrat
- Pentagram
- Zestaw pentagramów
- Okrąg
- Obracany kwadrat
- Kafelki
- Gwiazda
- Batarang
- Róża
- Rozeta
- Labirynt
- Galaktyka
- Ciastko

proGRAmowanie
- Gra w “20”
- Polowanie na kaczki
- Labirynt
- Tabliczka mnożenia dla Ani

Zbiór zadań
- Grupa 0
- Grupa A
- Grupa B
- Grupa C

Objaśnienia
O autorze
Przydatne linki

Wstęp

Programowanie w szkole

Zgodnie z Podstawą Programową, *każdy absolwent liceum, czy technikum powinien wykazywać się umiejętnością rozwiązywania problemów z użyciem komputera oraz stosowania algorytmicznego podejścia do zadań.*

Szkoła stara się kształcić tę umiejętności u uczniów już od pierwszych lat nauki. Niestety, często próbuje się w tym celu wykorzystywać tradycyjne, ale jednocześnie anachroniczne, środowiska programistyczne (np. Logo). Pochodzą one z zamierzchłych epok programistycznych, więc nie są dla uczniów atrakcyjne. Często ich jedyną „zaletą" jest fakt, że nauczyciele znają je i potrafią stosować w nauczaniu. Jednocześnie nie znają nowszych rozwiązań o podobnych funkcjach. Jest to wielka strata, biorąc pod uwagę znaczący wpływ nauki programowania na umiejętności kształtowane u uczniów. Młodzież mająca styczność z programowaniem notuje bowiem zwiększony stopień pomysłowości, kreatywności i dbałości o szczegóły oraz wykazuje się zróżnicowaniem pomysłów.

Optymalnym środowiskiem do nauki programowania byłoby takie, które cechowałaby przyjazność, prostota obsługi, bezpłatność, wieloplatformowość[1], a przy tym nowoczesność. Ideałem byłoby, gdyby zostało opracowane z myślą o edukacji.

```
 8  bool prim(int n)
 9  {//Czy liczba jest pierwsza? - Prawda albo fałsz
10     int k;
11     bool wynik;
12     if(n<=2)
13     {
14         if(n==1)
15         wynik=0;
16         if(n==2)
17         wynik=1;
18     }
19     else
20     {
21         for(k=2; (k*k<=n)&&(n%k!=0);k++)
22         {}
23         if(n%k!=0)
24         wynik=1;
25         else
26         wynik=0;
27     }
28     return wynik;
29  }
30
```

To prosty skrypt w C++. Czy jest też prosty dla uczniów?

[1] Możliwość pracy na wielu systemach operacyjnych

O Scratchu

Istnieje środowisko, które nie tylko spełnia te cechy, ale także posiada wiele innych pożądanych z punktu widzenia edukacji. Jest nim Scratch, stworzony przez Massachusetts Institute of Technology, opracowany specjalnie dla edukacji przez znany zespół Mitchela Resnicka "Lifelong Kindergarten" - pracownię MIT Media Lab. Scratch pozwala na proste tworzenie aplikacji mogących wykorzystywać dźwięki, rysować, porównywać kolory i wiele więcej. Jest całkowicie bezpłatny i działa na różnych systemach operacyjnych, co czyni go dostępnym dla każdego ucznia. Jedynym wymaganiem systemowym dla niego jest posiadanie dowolnej przeglądarki internetowej oraz dostępu do Internetu. Scratcha z powodzeniem można już używać w pracy z sześciolatkami. Ciekawym jest również fakt, że aplikacje w nim stworzone można łatwo udostępnić innym użytkownikom, czy umieścić na dowolnej stronie internetowej lub blogu.



Mitchel Resnick - ojciec Scratcha we własnej osobie

Aplikacje w Scratchu buduje się (dosłownie) w edytorze bloków funkcjonalnych. Nie wymaga to wcześniejszej znajomości żadnego języka programowania. Wszystkie komendy wypisane są na bloczkach, w języku polskim. Mają one, w zależności od swojego przeznaczenia, różne kolory. Dzięki temu struktura tworzonej aplikacji znacznie zyskuje na przejrzystości.

Warto zastanowić się nad wykorzystaniem Scratcha w szkole. Ze względu na nowe podejście do sposobu tworzenia aplikacji, ma on duży potencjał dydaktyczny. Jasna i przemyślana

struktura programu w edytorze bloczków znakomicie wspomaga uczenie myślenia algorytmicznego.

Opinia studentów o Scratchu

Wśród studentów Matematyki z Informatyką została przeprowadzona ankieta na temat Scratcha. Studenci, po odbyciu sześciogodzinnego kursu algorytmiki opartym o to środowisko mieli za zadanie odpowiedzieć na kilka pytań.

Analizując odpowiedzi respondentów można zauważyć, że każdy z nich zadeklarował, iż wybrałby Scratcha, gdyby miał przeprowadzać zajęcia z algorytmiki. Studenci jednogłośnie uznali również, że w tym środowisku programuje się łatwiej, aniżeli w innych (np. Logo, Pascal). Te opinie można uznać za reprezentatywne, gdyż blisko 2/3 ankietowanych miało wcześniej styczność z programowaniem.

Studenci zostali również poproszeni o to, by każdy z nich wypisał trzy wady i zalety Scratcha, jakie udało im się zauważyć, podczas kursu algorytmiki. Najczęściej powtarzającymi się zaletami był przyjazny interfejs, łatwość edycji oraz bezpłatność. Respondenci mieli problem ze znalezieniem wad Scratcha. Wśród nich został wymieniony brak możliwości dowolnej zmiany rozmiaru okna sceny (dostępne są dwa wbudowane ustawienia).

Szczegółowe wyniki ankiety dostępne są do obejrzenia w serwisie eNnauczanie.com.

Scratch a Logo: Porównanie kryterialne

	Scratch	Logo
Interfejs w języku polskim	Tak	Tak
Nauka komend “na pamięć”	Nie	Tak
Możliwość tworzenia rysunków	Tak	Tak
Instrukcje warunkowe	Tak	Tak
Pętle	Tak	Tak
Tworzenie własnych procedur	Tak	Tak
Współpraca z bazą danych	Tak	Nie
Programowalne elementy	Dowolna liczba duszków	Jeden żółw
Proste dodawanie dźwięku do aplikacji	Tak	Nie
Tworzenie animacji	Tak	Nie

Publikowanie dźwięku w aplikacji	Tak	Nie
Wieloplatformowość	Tak	Nie
Wsparcie użytkowników i nauczycieli	Duże	Małe

Przygotowanie do pracy

Co nowego w Scratchu 2.0?

Scratch w wersji 2.0 wprowadził wiele nowości w stosunku do swojego poprzednika. Najważniejszymi i najbardziej zauważalnymi z nich są:

- praca wyłącznie w chmurze – Scratch już nie wymaga instalacji na komputerze, a wszystkie projekty są przechowywane na koncie użytkownika,
- możliwość tworzenia własnych klocków, wykonujących żądane przez nas czynności (np. rysowanie kwadratu o danym boku),
- podgląd kodu programu innego użytkownika bezpośrednio w przeglądarce - jeżeli chcemy wiedzieć jak inny użytkownik rozwiązał pewnien problem lub doszedł do jakiegoś efektu, możemy podejrzeć wnętrze jego programu,
- zmienne globalne, dzięki którym można np. tworzyć gry zapisujące najwyższe wyniki użytkowników,
- praca na obiektach wektorowych, dzięki czemu część graficzna programu wyglądają o wiele ładniej.



Zaletą nowego Scratcha jest to, że poprawki i udoskonalenia do programu będą wprowadzane do niego w sposób niezauważalny dla użytkownika. Koniec ze sprawdzaniem strony programu, czy pojawiła się już do niego aktualizacja. Włączając stronę scratch.mit.edu zawsze przed naszymi oczami ujrzymy najnowszą wersję Scratcha.

Zakładanie konta w serwisie Scratch

Scratch jest dostępny dla każdego komputera posiadającego przeglądarkę internetową oraz dostęp do Internetu. Od wersji 2.0 działa on bowiem wyłącznie w chmurze. Mamy więc dostęp do swoich projektów z każdego miejsca na ziemi. Wystarczy założyć konto w serwisie Scratcha.

By to zrobić:

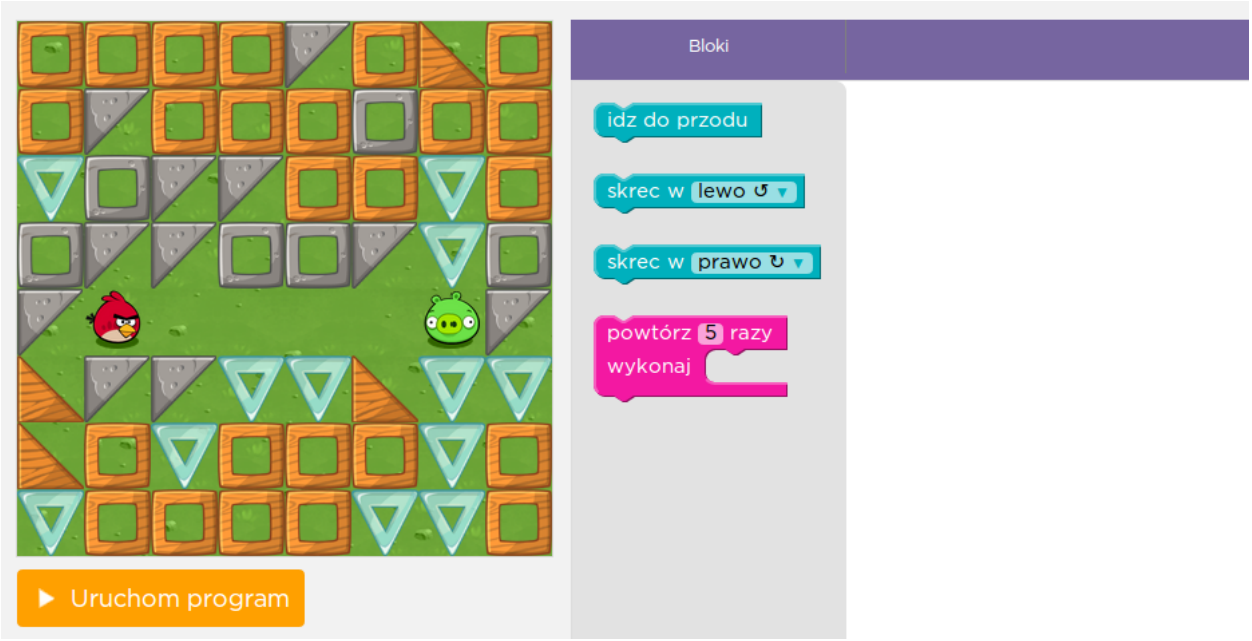
1. Włącz stronę scratch.mit.edu
2. Kliknij na przycisk **Dołącz do Scratch**, widoczny u góry strony.
3. W okienku, które się pojawi, wybierz nazwę użytkownika oraz hasło, które będzie pozwalało Ci się zalogować w serwisie.

4. Kolejnym krokiem będzie podanie kilku informacji o sobie, czyli daty urodzin, płci, państwa z którego się pochodzi oraz aktualnego adresu e-mail. Będą na niego przychodziły powiadomienia z serwisu, a w razie problemów z zalogowaniem, będzie można zresetować z jego pomocą hasło.

Rozgrzewka

Przed przystąpieniem do pracy ze Scratchem warto poczynić pewne przygotowania. Nieważne, czy w zajęciach występujemy jako prowadzący, czy uczestnicy - przyda się niewielka rozgrzewka. Istnieje fantastyczny serwis, który doskonale nadaje się na zajęcia będące wstępem do programowania w Scratchu. To code.org.

Serwis zawiera w sobie kurs przygotowawczy przeznaczony dla uczniów szkół podstawowych, stworzony w języku polskim. Można go włączyć bezpośrednio, wpisując adres learn.code.org. Zadania w nim są banalnie proste oraz są wspomagane dużą liczbą materiałów dodatkowych, objaśnień i porad eksperckich, udzielanych przez specjalistów takich jak Mark Zuckerberg[2].



Kurs przygotowujący do nauki programowania na stronie code.org

Warto zarejestrować się w tymże serwisie, gdyż w ten sposób osiągane wyniki będą się zapisywały na naszym profilu. Ukończenie odpowiednich sekcji kursu będzie skutkowało otrzymaniem certifikatów potwierdzających umiejętności.

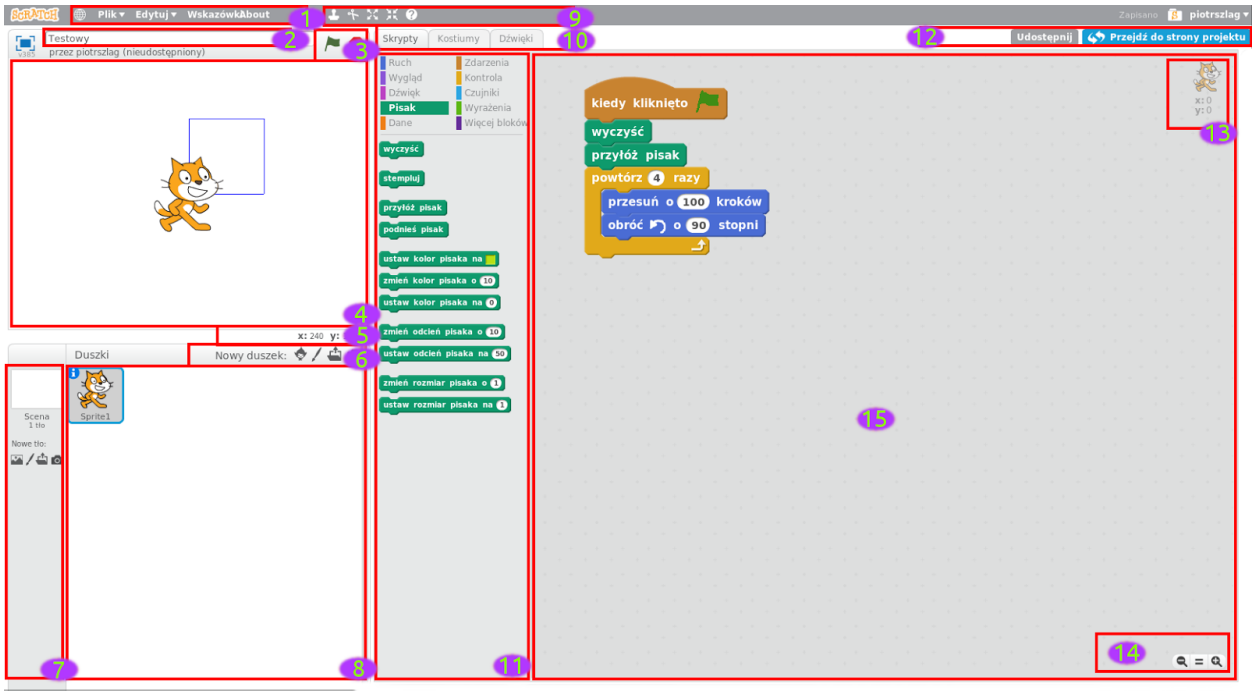
[2] Twórca Facebooka

Pierwsze kroki ze Scratchem

W tym dziale poznasz podstawy obsługi programu Scratch. Dowiesz się, co oznaczają poszczególne elementy okna aplikacji. Stworzysz również swoje pierwsze projekty, które pomimo swojej prostoty będą przedstawiały ciekawe efekty.

Okno programu

Główne okno Scratcha jest podzielone na kilka wyróżniających się sekcji, które nawzajem ze sobą współpracują. Znajomość ich funkcji jest kluczowa do sprawnego poruszania się w programie, lecz zaznajomienie się z nią nie sprawi Ci większej trudności. Interfejs programu został bowiem tak zaprojektowany, by radzili z nim sobie nawet sześciolatkowie.



W opisywaniu poszczególnych sekcji głównego okna Scratcha będziemy się odwoływać do powyższego rysunku, na którym najważniejsze z nich zostały zaznaczone i ponumerowane.

I tak:

1. Główne menu programu. Tu możesz program zapisać, pobrać na swój komputer, czy też wgrać projekt aplikacji stworzonej w starszej wersji Scratcha. Jeśli potrzebujesz więcej miejsca na kod swojego programu, przełączysz się w **Układ małej sceny**. Klikając na napis **Wskazówki**, z prawej strony rozwinie się menu szybkiej pomocy. Pierwszy przycisk z lewej strony pozwoli Ci zmienić język interfejsu programu.
2. Tu wybierzesz nazwę dla swojego programu.
3. Uruchamianie i zatrzymywanie skryptu dla wszystkich duszków. Zieloną flagą uruchamisz skrypt[3], czerwonym znakiem **stop** zahamujesz działanie całego programu.
4. Scena – tu są wyświetlane duszki wykonujące zaprogramowane przez Ciebie instrukcje.

[3] W kodzie programu musi zostać umieszczony bloczek z zieloną flagą

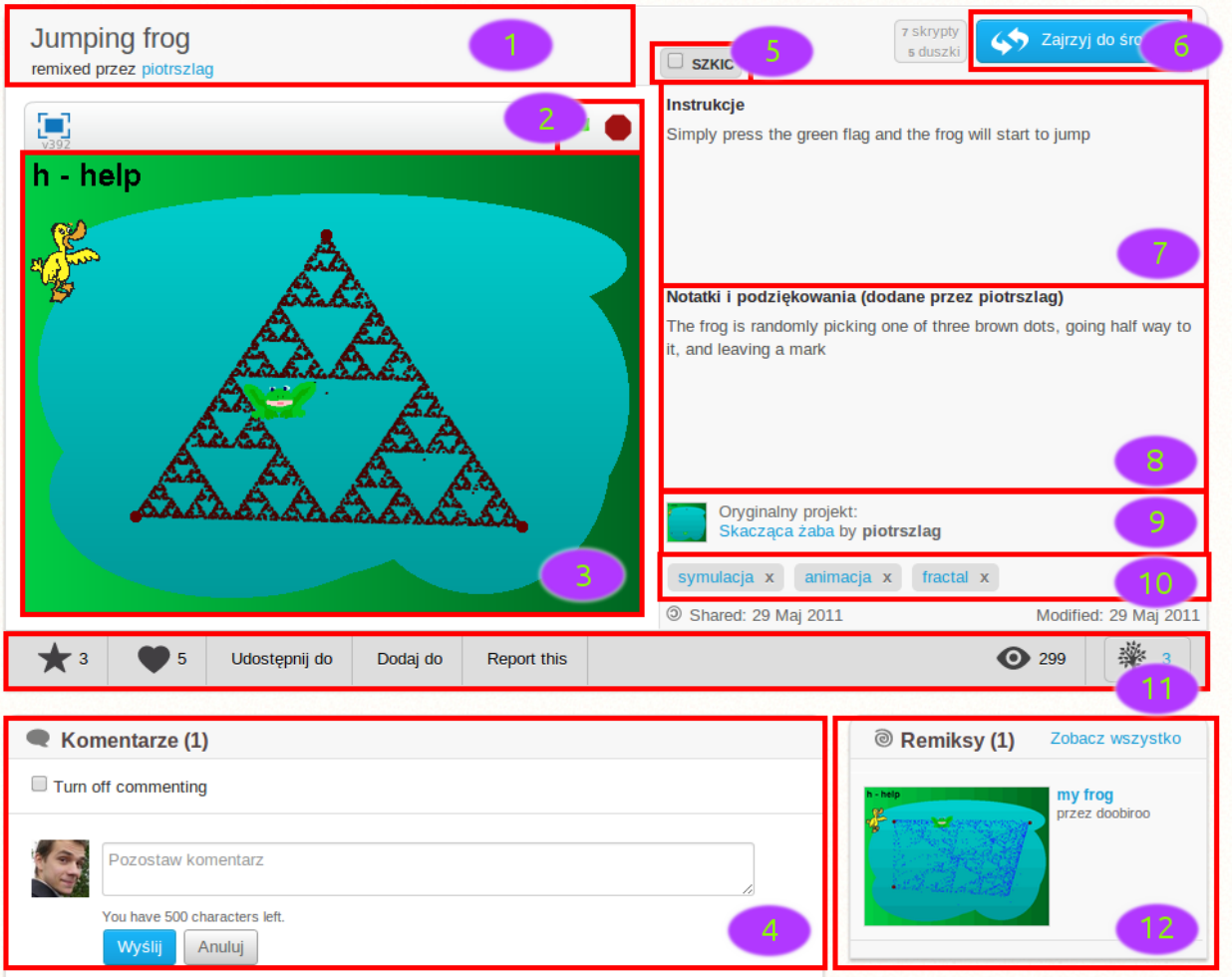
5. Ten obszar podaje Ci aktualne współrzędne miejsca, w którym jest umieszczony kursur myszki.
6. Guziki (od lewej) pozwalają odpowiednio na wybranie duszka z bazy Scratcha, namalowanie własnego duszka we wbudowanym edytorze programu, wgranie duszka z pliku, który posiadasz na swoim komputerze oraz stworzenie duszka przy pomocy Twojej kamerki internetowej.
7. Wybieranie sceny (czyli tła) dla wykonywanego skryptu. Domyślnie jest to biała plansza. Podobnie jak to miało miejsce powyżej nowe sceny możesz wgrywać z bazy, którą oferuje Scratch, rysować samemu, wgyrywać ze stworzonych przez siebie plików, czy też sfotografować swoją kamerką internetową. Scenę również możesz zaprogramować np. żeby zmieniała się na inną po pięciu sekundach. Wystarczy, że na nią klikniesz (pojawi się wtedy wokół niej niebieska ramka), a następnie umieścisz odpowiednie bloczki w sekcji skryptów.
8. Wybieranie duszka, którego w danym momencie chcesz programować. Pamiętaj, że Twój program może mieć kilka duszków, z czego każdy może być inaczej zaprogramowany. Sekcja skryptów będzie się zmieniała w zależności od wybranego w danym momencie duszka - poznasz go po tym, że jest otoczony niebieską ramką. Jeśli chcesz usunąć któregoś duszka, kliknij na nim prawym przyciskiem myszki, a następnie wybierz opcję *Usuń*. W ten sam sposób możesz go zduplikować, albo ukryć.
9. Narzędzia pozwalające na szybką edycję duszka. Od lewej jest to duplikowanie duszka, wycinanie, powiększanie i jego pomniejszanie. Wystarczy wybrać narzędzie, a następnie kliknąć na wybranego duszka widocznego w obszarze wykonywania programu.
10. Zakładki pozwalające na zmianę zawartości sekcji, która domyślnie wyświetla skrypty. Możesz tam, klikając na napis *Kostiumy/Tła*[4] dokonywać zmiany w wyglądzie swoich duszków, czy scen. Zakładka *Dźwięki* pozwoli Ci na dodanie do swojego projektu odgłosów, które następnie mogą być wykorzystywane w działaniu programu np. Kot będzie mówił “Auć”, gdy będzie się uderzał o ścianę.
11. Bloczki potrzebne do stworzenia programu. Są one pogrupowane kolorystycznie w zależności od funkcji, jaką spełniają (niebieskie – ruch, różowe – dźwięk itd.). Klikając prawym przyciskiem na poszczególnym bloczku, możesz wywołać okno pomocy, szczegółowo wyjaśniające jego przeznaczenie.
12. Tymi guzikami możesz udostępnić swój projekt tak, by każda osoba na świecie mogła go podziwiać. Klikając na przycisk *Przejdź do strony projektu*, będziesz mógł podejrzeć stronę, którą zobaczą inni użytkownicy, po udostępnieniu przez Ciebie projektu. Zostanie ona szerzej omówiona w następnej sekcji.
13. Ten obszar informuje o współrzędnych, które określają położenie aktywnego w danym momencie duszka.
14. Widocznymi guzikami możesz (od lewej) pomniejszyć, powrócić do ustawień domyślnych lub powiększyć bloczki Twojego projektu.

[4] Tekst ten zmienia się w zależności, czy w danech chwili otoczony niebieską ramką jest duszek, czy scena.

15. Skrypty – w to miejsce należy przeciągać bloczki, by stworzyć swój program. Możesz łączyć ze sobą dowolną liczbę pasujących do siebie bloczków, tworząc ich segmenty. Bloczki łączą się, gdy podczas ich zbliżania ukazuje się między nimi biały pasek. Usuwanie niepotrzebnych bloczków polega na kliknięciu na nim prawym przyciskiem myszki i wybraniu opcji *Usuń*. W ten sam sposób możesz je duplikować.

Strona projektu

Gdy już skończysz pracę ze swoim programem, z pewnością będziesz chciał się nim pochwalić przed szerszą publicznością. Do tych celów służy właśnie *Strona projektu*. To w tym miejscu opisujesz swój projekt, a inni użytkownicy mogą go podejrzeć lub wyrazić o nim swoje zdanie. Im lepiej opisany program, tym więcej osób będzie go znajdywało.



Zajmiemy się teraz opisywaniem poszczególnych elementów *Strony projektu*. Będziemy się odwoływać do powyższegi rysunku, na którym oznaczono i ponumerowano najważniejsze sekcje.

1. Kilkając w to miejsce będziesz mógł szybko zmienić tytuł swojego projektu. Jeżeli Twoja praca powstała na podstawie projektu innego użytkownika, to będzie tam wyświetlany jego login.
2. Guziki którymi możesz uruchomić program (zielona flaga) lub go zatrzymać (czerwony przycisk stop).
3. Ekran, w którym program jest wykonywany. To tu dzieją się najbardziej interesujące rzeczy.
4. W tym miejscu użytkownicy, którzy będą oglądali Twoją pracę, będą mogli zostawiać komentarze. Jeśli chcesz, to zaznaczając widoczne w tym sektorze pole możesz wyłączyć możliwość komentowania.
5. Klikając na to pole możesz pokazać innym użytkownikom, że pomimo opublikowania projektu, nie jest to jego ostateczna wersja - czyli jest tzw. szkicem.
6. Guzik pozwala na zajrzenie do kodu wyświetlanego w danym momencie projektu. Jest to szczególnie przydatne, gdy oglądasz ciekawy projekt i chciałbyś zobaczyć, jak został on stworzony.
7. To pole jest bardzo ważne do osób odwiedzających Twoje projekty. Powinieneś wpisać tu niezbędne instrukcje dotyczące działania Twojego projektu tak, by nikt nie miał problemów z jego używaniem.
8. Tu wpisz notatki dotyczące projektu. Mogą to być Twoje plany na rozwój aplikacji lub zwyczajne podziękowania za pomoc.
9. Jeżeli do stworzenia aplikacji została użyta praca innegu użytkownika, to właśnie w tym miejscu będzie się znajdywała informacja o tym.
10. W tym polu możesz zapisać tagi związane z Twoim projektem. Jeżeli robisz kalkulator, to takim tagiem może być np. *matematyka*. Im więcej tagów pasujących do Twojej pracy, tym więcej ludzi odwiedzi tę stronę, bo będzie ona zwyczajnie lepiej wyszukiwalna.
11. Szary panel będzie Cię informował o tym jak inni użytkownicy na niego reagują. Poszczególne przyciski oznaczają (od lewej):
    a. polubienie projektu,
    b. ukochanie projektu[5],
    c. udostępnienie projektu - możesz stąd pobrać kod pozwalający na osadzenie programu na swojej stronie czy też plik programu, który otworzysz w starszej wersji Scratcha,
    d. dodanie projektu do swojego studia - takiej playlisty programów,
    e. zgłoszenie projektu jako łamiącego regulamin serwisu,
    f. liczbę wszystkich wyświetleń,
    g. drzewo remiksów projektu, to miejsce gdzie możesz zobaczyć jak Twój projekt rozrastał się, poprzez przekształcanie go przez innych użytkowników,
12. Sekcja pokazuje jakie remiksy udało się stworzyć innym użytkownikom, na bazie obecnie oglądanego projektu.

[5] Swoją drogą to ciekawe, że Scratch rozróżnia lubienie od kochania :)

Zaczynamy programować

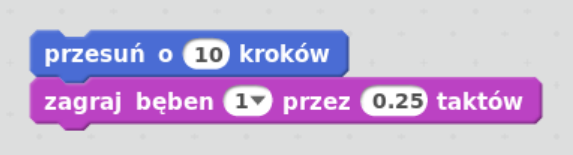
Samo słowo *Programowanie* przyprawia niektórych o gęsią skórkę i odrzuca ich jak najdalej od komputera. Niewiadomo czemu ta prosta czynność kojarzy się wielu osobom z zajęciem, którym mogą zajmować się jedynie osoby szczególnie uzdolnione. Za chwię sam przekonasz się, że to nic trudnego oraz, że może to być całkiem dobrą zabawą.

Tworzymy pierwszy program

1. Przeciągnij bloczek **przesuń o ... kroków** do sekcji *Skryptów*. Kliknij na bloczek. Kot się poruszy.



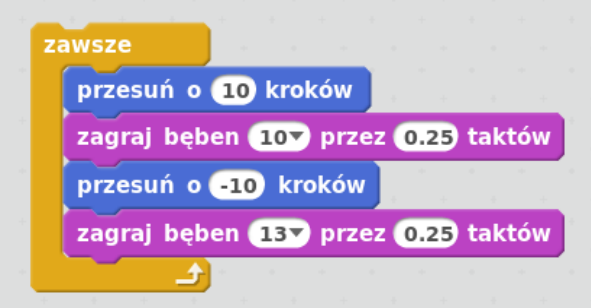
2. Przeciągnij bloczek **zagraj bęben** (możesz go znaleźć zakładce *Dźwięk*) i połącz go z **przesuń o ... kroków**.



3. Z listy rozwijalnej bloczku **zagraj bęben** możesz wybrać wiele ciekawych dźwięków. Wypróbuj je.
4. Dodaj kolejny bloczek **przesuń o ... kroków**, kliknij na okienko wewnątrz niego i wstaw przed liczbą znak minus. Następnie dodaj kolejny bloczek **zagraj bęben** i wybierz inny instrument. Kliknij na którymkolwiek z bloczków, żeby uruchomić program.



5. Przeciągnij bloczek **zawsze** (zakładka *Kontrola*) i upuść go na szczycie instrukcji. Powinien otaczać wszystkie dodane do tej pory bloczki.



6. Kliknij na jednym z bloczków by uruchomić program.
7. Żeby przerwać działanie skryptu, naciśnij na czerwony znak stop w prawym górnym rogu programu.
8. Wybierz bloczek **kiedy kliknięto** i umieść go na szczycie listy instrukcji. Od teraz, zawsze gdy klikniesz zieloną flagę w prawym górnym rogu programu, zostanie on



uruchomiony. Żeby zatrzymać aplikację, kliknij na znak stop.
9. Przeciągnij bloczek **zmień efekt** (zakładka *Wygląd*). Nie dodawaj go do istniejącej już listy instrukcji, ale ustaw go obok. Kliknij na bloczek i zobacz co stanie się z kotem.



10. Dodaj bloczek kiedy **kiedy klawisz ... naciśnięty**. Ułóż go na bloczku **zmień efekt**, a następnie naciśnij *Spację* na swojej klawiaturze (z listy rozwijalnej tego bloczka możesz również wybrać inny klawisz).

Gratulacje! Właśnie skończyłeś tworzyć swój pierwszy program w Scratchu. Zauważ, że powstałe dwie instrukcje mogą działać niezależnie od siebie tzn. po włączeniu zielonej flagi kot zacznie tańczyć i dodatkowo za każdym razem, gdy zostanie wciśnięta spacja, zmieni on swój kolor. Na tym krótkim przykładzie widać również, że skrypt programu jest czytelny i zrozumiały nawet dla osób niemających wcześniej kontaktu z programowaniem.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/zaczynamy-programowac/tworzymy-pierwszy-program

Kalejdoskop

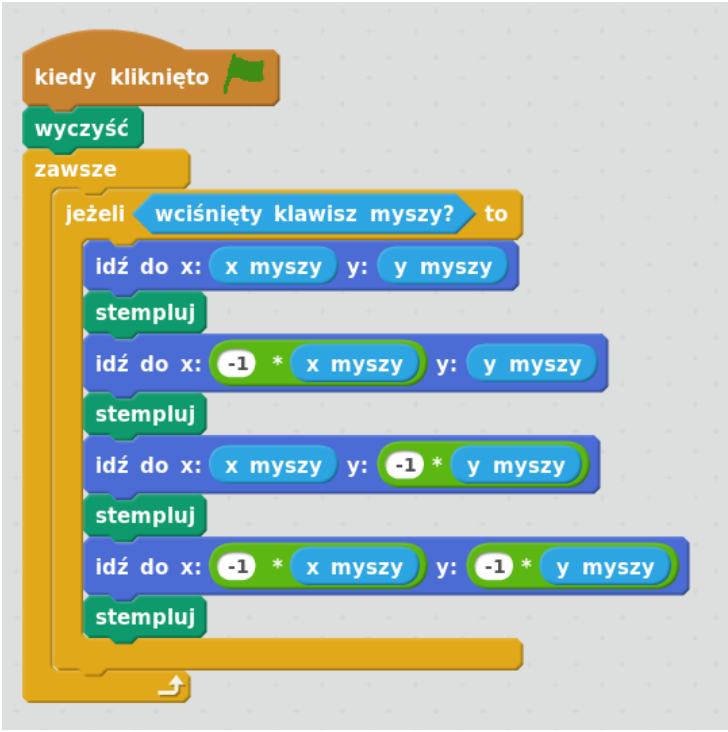
1. Uruchom nowy projekt w Scratchu. Przejdź do zakładki *Kostiumy*.
2. Pojawi się nowa sekcja, służąca do edycji wyglądu duszków. Całkowicie wymaż kota. Możesz do tego użyć guzika *Wyczyść*. Gdy to zrobisz, to przy pomocy narzędzia *Elipsy* narysuj czerwone kółko[6]. Kółko powinno być ciut większe niż kursor myszki. Naciśnij na guzik *OK*, gdy skończysz.
3. Wróć teraz do zakładki *Skrypty* i złóż z bloczków kod widoczny na rysunku poniżej. Zamieszczone instrukcje działają w ten sposób, że w momencie naciśnięcia zielonej

[6] Idealne koło stworzysz, gdy w momencie rysowania wciśniesz klawisz **Shift**



flagi następuje czyszczenie ekranu, po czym startuje niekończąca się pętla wykrywająca kliknięcia myszki w obszarze sceny. Gdy takie nastąpi, to na planszy, w miejscu kursora, zostanie postawione stworzone wcześniej czerwone kółko. Naciśnij zieloną flagę i postaraj się coś namalować. Gdy będziesz chciał skończyć, kliknij na czerwony znak stop.

4. Dodaj do poprzednich instrukcji kilka nowych bloczków i ponownie naciśnij zieloną flagę. Zobacz, jaki efekt udało ci się uzyskać.



5. Następna edycja twojego skryptu będzie polegała na dodaniu tylko jednego bloczka, ale spowoduje dość duże zmiany wizualne. Z zakładki *Wygląd* dodaj w dowolne miejsce w środku pętli **zawsze, jeżeli ...** bloczek **zmień efekt kolor o ...** i wstaw tam dowolną wartość. Kliknij na zieloną flagę i zobacz, co udało ci się osiągnąć.



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/zaczynamy-programowac/kalejdoskop

Obliczanie pola kwadratu

Zajmiesz się teraz tworzeniem prostego programu, którego zadaniem będzie obliczanie pola kwadratu o długości boku podanej przez użytkownika obsługującego twoją aplikację. Poznasz również bloczki **zapytaj i czekaj** oraz **odpowiedz**, które są jednymi z najważniejszych, gdyż pozwalają na pobieranie informacji z zewnątrz programu i ich dalsze przetwarzanie.



Uruchom Scratcha i ułóż w nim ciąg instrukcji identyczny, jak ten na rysunku powyżej. Gdy tego dokonasz, wciśnij zieloną flagę. Kot spyta się o bok kwadratu i będzie czekał na wpisanie przez ciebie wartości do pola tekstowego widocznego w dolnej części sceny. Wpisz tam dowolną liczbę i zatwierdź klawiszem Enter. Po chwili kot powie ile wynosi pole kwadratu o długości boku podanej przez ciebie.

Twój program już poprawnie oblicza pole kwadratu, ale nie wie jeszcze o tym, że bok figury



geometrycznej musi być liczbą dodatnią. Zmodyfikuj swoją listę instrukcji na taką, jaka jest zamieszczona na rysunku powyżej. Skrypt działa teraz tak, że po podaniu przez użytkownika wartości boku kwadratu następuje sprawdzenie przez program, czy wartość napisana przez użytkownika jest liczbą dodatnią. Jeżeli tak jest, to program przechodzi do obliczania pola kwadratu. Jeżeli nie, przypomina użytkownikowi, jakie warunki musi spełniać bok figury geometrycznej i prosi o ponowne podanie tej wartości - aż do skutku.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/zaczynamy-programowac/obliczanie-pola-kwadratu

Zmienne i listy

Za chwilę poznasz bardzo ważne narzędzia, wykorzystywane przez każdego programistę. Wszystkie poważne programy muszą bowiem operować na danych, a więc języki programowania potrzebują mechanizmy, by je gdzieś przechowywać. Do tego właśnie w Scratchu służą Zmienne i Listy.

Zmienne

Gdy myślisz o zmiennych, możesz wyobrażać sobie je jako pojemniki, które pozwolą Ci przechowywać różne przedmioty. Te przedmioty to tak naprawdę informacje, takie jak liczby i tekst. Możesz je wykorzystywać w swoich programach na wiele sposobów - wypisywać, porównywać, czy dokonywać na nich operacji matematycznych. Brzmi skomplikowanie? Gwarantuję, że nie jest. Zobaczysz to zaraz na prostym przykładzie.

Zajmiesz się teraz tworzeniem skryptu obliczającego pole prostokąta o bokach, których długość będzie podawana przez użytkownika. Dzięki temu poznasz w jaki sposób działają zmienne i jak je wykorzystywać w późniejszych projektach.
Uruchom Scratcha i tak jak to miało miejsce w programie obliczającym pole kwadratu, zapytaj użytkownika by podał jeden z boków prostokąta. Odpowiedź użytkownika będzie



przechowywana w klocku **odpowiedz**. Zauważ jednak, że gdybyśmy teraz chcieli zapytać o drugi bok prostokąta, to nowa odpowiedź użytkownika nadpisze tą pierwszą. Musisz zatem zapisać obie odpowiedzi tak, by potem móc z nich korzystać.

Stwórz dwie zmienne o nazwach *bok 1* i *bok 2*. Zrobisz to przechodząc w sekcji bloczków do zakładki **Dane** i klikając na guzik **Stwórz zmienną**. Zauważ, że możesz tworzyć zmienne dostępne tylko dla poszczególnych duszków, albo takie, z których korzystać mogą wszystkie jednocześnie. Jeżeli wszystko wykonałeś poprawnie, powinieneś w zakładce **Dane** widzieć nowe klocki, właśnie o takich nazwach.

Jeżeli będziesz teraz chciał, by wartości podawane przez użytkownika były zapamiętywane na dłuższy czas, użyj kombinacji bloczków **ustaw ... na ...** i **odpowiedz**. Gotowy kod programu obliczającego pole prostokata możesz podejrzeć na rysunku powyżej

Jeżeli chciałbyś ukryć wartości zmiennych, które są pokazywane w oknie wykonywania programu - odhacz je w sekcji **Danych**.

Gratulacje. Potrafisz już pracować na zmiennych. Teraz czas na listy.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/zaczynamy-programowac/pole-prostokata

Listy

Listy w Scratchu[7] to narzędzie, które pozwala Ci na przechowywanie wielu danych w jednym miejscu. Możesz patrzyć się na nie jak na zmienną, która zawiera w sobie wiele innych zmiennych. Czy jest ograniczenie liczby pozycji w liście? Nie. Możesz ich umieścić tam dowolną ilość. Każda z wartości będzie wtedy przypisana do konkretnego miejsca na liście.

[7] W innych językach programowania nazywane jako *Tablice,* stanowią rozwinięcie idei zmiennych



Listy można wykorzystywać w bardzo ciekawy sposób. By to pokazać, spróbujemy razem zrobić program przepowiadający przyszłość - coś w rodzaju Horoskopu. Stworzymy wpierw trzy listy[8]: *Miłość*, *Pieniądze* i *Zdrowie*. Teraz do każdej z list wpisz po trzy przepowiednie[9] za pomocą bloczka **dodaj ... do …** . Zwracaj uwagę, czy dodajesz dobre przepowiednie do dobrej listy. Po tej czynności twój skrypt powinien wyglądać podobnie do tego na rysunku powyżej.

Jeżeli masz już bazę przepowiedni, pora zaprogramować Twojego duszka tak, aby z każdej listy losował po jednym zdaniu do danej kategorii. Osiągniemy to, korzystając z bloczka **losuj od ... do ...** oraz dodania nowego skryptu, który będzie działał równolegle do tego, który stworzyłeś powyżej. Zauważ w tym miejscu, że nie ma żadnego ograniczenia co do liczby skryptów, które będą rozpoczynały swoje działanie w momencie kliknięcia zielonej flagi.

Obok skryptu z rysunku powyżej ułóż kolejny, taki jak ten widoczny na rysunku poniżej.



[8] Robi się to tak samo jak tworzenie Zmiennych
[9] Jeżeli chcesz, możesz później rozbudować swój program, dodając więcej wróżb

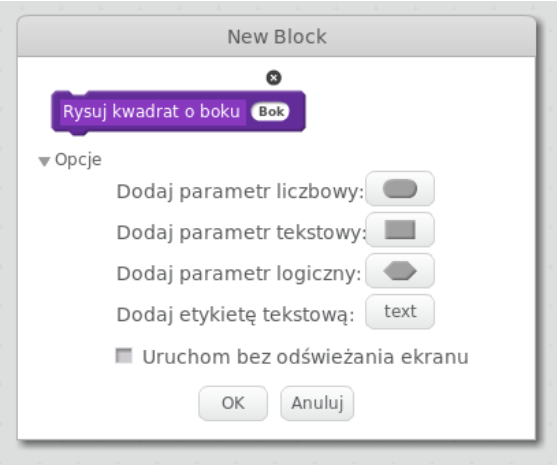
Zobacz działanie programu:
enauczanie.com/metodyka/scratch/zaczynamy-programowac/horoskop

Tworzenie własnych bloczków

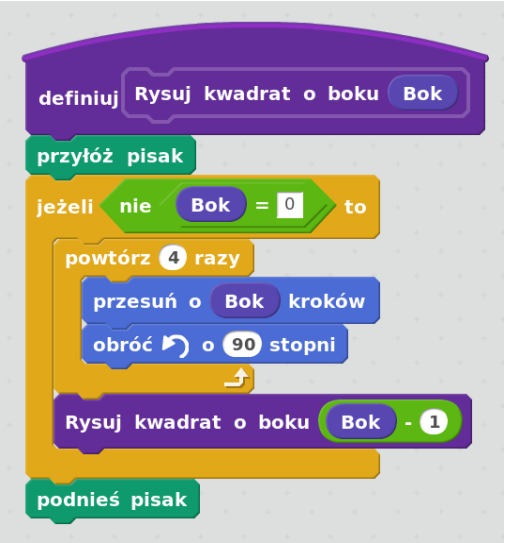
Przyjdzie czas, że będziesz chciał w Scratchu tworzyć coraz to bardziej zaawansowane programy. Jak nietrudno wywnioskować - im bardziej rozbudowany kod będziemy chcieli stworzyć, tym więcej klocków będzie należało użyć. Poznasz teraz metodę na przyspieszenie prac w bardziej zaawansowanych projektach.

Od wersji 2.0, Scratch pozwala na tworzenie swoich własnych bloczków. Jeżeli miałeś już coś wcześniej wspólnego z programowaniem, to możesz utożsamiać je z procedurami. Jeżeli nie, to nie martw się - sam zobaczysz, że to nic trudnego. Dzięki tej nowej funkcji możesz np. utworzyć bloczek, który będzie rysował kwadrat o danym boku. Wtedy, zawsze gdy będziesz chciał narysować kwadrat użyjesz tylko jednego bloczka, zamiast układać potrzebny kod.

Tyle teorii. Teraz zobaczysz, jak można wykorzystywać własne bloczki w praktyce. Naszym celem będzie stworzenie bloczka, który będzie rysował kwadrat o podanym przez użytkownika boku. Kwadrat ten będzie zamalowany w środku. W oknie tworzenia programu w Scratchu, przejdź do zakładki *Więcej bloczków*, a następnie na *Stwórz blok*. Pojawi się nowe okienku, gdzie będziesz projektował wygląd swojego bloczka. W pierwsze pole wpisz *Rysuj kwadrat o boku* oraz wstaw *Parametr liczbowy* i zmień jego nazwę na *bok*. Powinieneś osiągnąć taki sam efekt jak ten widoczny na rysunku poniżej.



Nadeszła pora na pokazanie Scratchowi, jak ma rysować kwadraty wypełnione w środku. Z pewnością zauważyłeś nowy, fioletowy bloczek, który pojawił się w sekcji *Skryptów*. Dołączając do niego kolejne fragmenty kodu sprawisz, by bloczek zachowywał się tak, jak będziemy tego chcieli. Umiesz rysować już kwadrat. Pewnie zastanawiasz się jak sprawić, żeby był on wypełniony kolorem . To bardzo proste. Twój duszek zwyczajnie musi narysować jeden kwadrat, potem drugi - mniejszy, potem kolejny - jeszcze mniejszy i tak dalej. Po uruchomieniu da to efekt wypełnionego kwadratu. Gotowy skrypt powinien wyglądać jak ten poniżej.



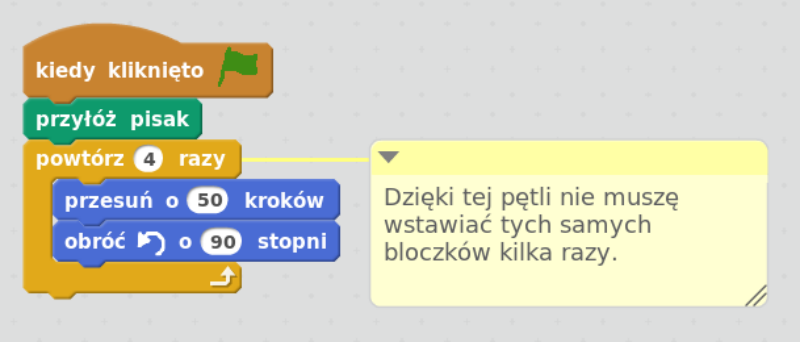
Po tej czynności, gdy będziesz chciał narysować kwadrat z wypełnionym środkiem, będziesz musiał wstawić w kodzie programu swój nowy bloczek. To ogromne udogodnienie, które wielokrotnie okaże się pomocne w Twojej przygodzie z programowaniem. Jeśli nie rozumiesz zasady działania powyższego kodu, przeczytaj koniecznie sekcję poświęconą Rekurencji, którą znajdziesz w rozdziale *Objaśnienia*.

Komentarze

Komentarze, wstawiane do skryptów, bardzo ułatwiają szybkie zorientowanie się w kodzie podczas jego odczytywania przez inne osoby. Mogą okazać się szczególnie przydatne, gdy twój program będzie zawierał wiele linijek kodu lub gdy otworzysz go po dłuższej przerwie i nie będziesz już pamiętał jakie czynności dany fragment kodu wykonywał. Ponieważ komentarze nie są bloczkami, aplikacja będzie je ignorowała.

Żeby dodać komentarz do swojego programu:

1. Kliknij prawym przyciskiem myszki obok polecenia lub skryptu, do którego chcesz dodać komentarz.
2. Z menu wybierz **Dodaj komentarz** i wpisz odpowiednią notkę.
3. Jeśli zbliżysz komentarz do bloczka, który jest opisywany, zostanie on z nim powiązany.



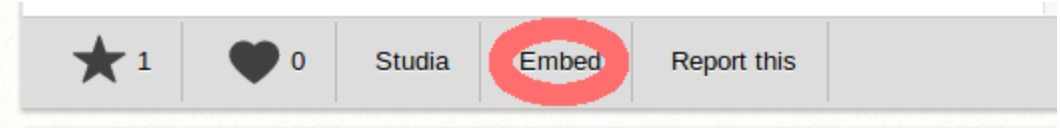
Tak wygląda wstawiony komentarz, przypisany do konkretnego bloczka

Publikowanie programów na stronach internetowych

Scratch pozwala na łatwe i szybkie umieszczanie stworzonych i opublikowanych aplikacji na swojej stronie www. Jest to tak proste, jak osadzenie obrazka, czy filmu YouTube.

Aby to zrobić:

1. Otwórz w przeglądarce program, który chcesz przenieść na swoją stronę.
2. Znajdź napis **Embed**, który znajduje się pod okienkiem z programem.
3. Zaznacz i skopiuj tekst widoczny w polu Embed.
4. Wklej ten kod w odpowiednim miejscu na swojej stronie www (by to zrobić będziesz prawdopodobnie musiał przejść do widoku html)



Rysowanie ze Scratchem

Sztandarowe programy tworzone w języku Logo (np. w aplikacji Logo Komeniusz), bazujące na rysowaniu dowolnych kształtów, możesz łatwo ułożyć w Scratchu. W ten sposób dalej będziesz tworzyć fantastyczne rysunki, zyskując jednocześnie przejrzystość i prostą w zrozumieniu listę instrukcji, którą oferuje Scratch. Dodatkowo, w przeciwieństwie do Logo, Scratch może rysować, wykorzystując do tego kilka duszków, z czego każdy z nich może być inaczej zaprogramowany. Znacznie zwiększa to pole do popisu dla uczniów, którzy chcą wykonać bardziej zaawansowany projekt.

Poniżej znajduje się zestaw rysunków stworzonych w Scratchu, wraz z instrukcjami potrzebnymi do ich samodzielnego wykonania. Nie krępuj się zmianą parametrów, czy przestawianiem bloczków. Możliwe, że w ten sposób stworzysz swój unikatowy rysunek, którym będziesz mógł się pochwalić przed innymi.

Kwadrat



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/kwadrat

Pentagram



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/pentagram

Zestaw pentagramów



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/zestaw-pentagramow

Okrąg



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/okrag

Obracany kwadrat



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/obracany-kwadrat

Kafelki



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/kafelki

Gwiazda



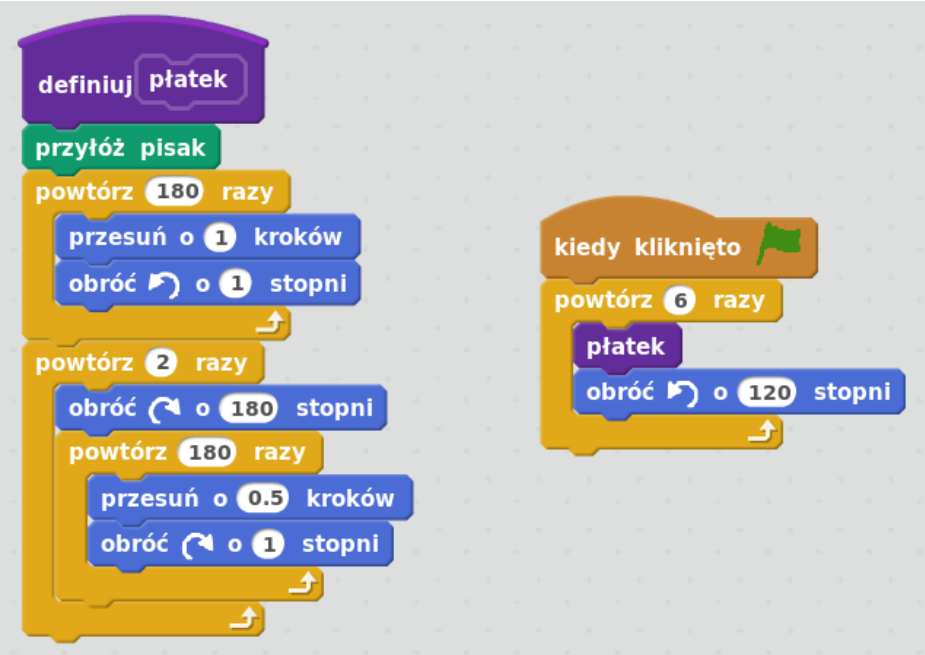
Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/gwiazda

Batarang



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/batarang

Róża



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/roza

Rozeta



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/rozeta

Labirynt



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/labirynt

Galaktyka



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/galaktyka

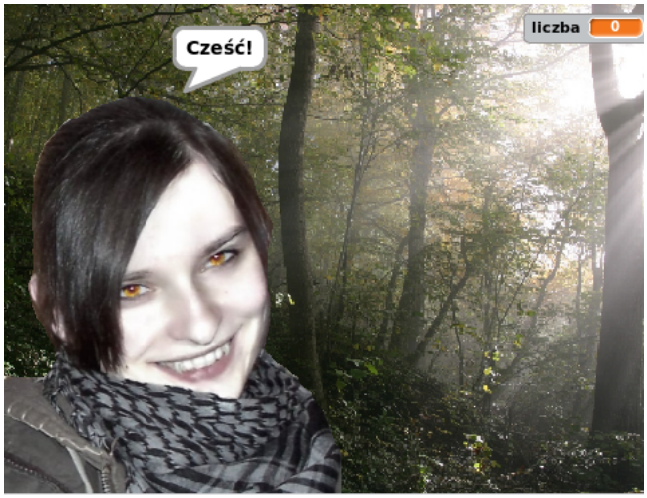
Ciastko



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/rysowanie-ze-scratchem/ciastko

proGRAmowanie

Gra w “20”

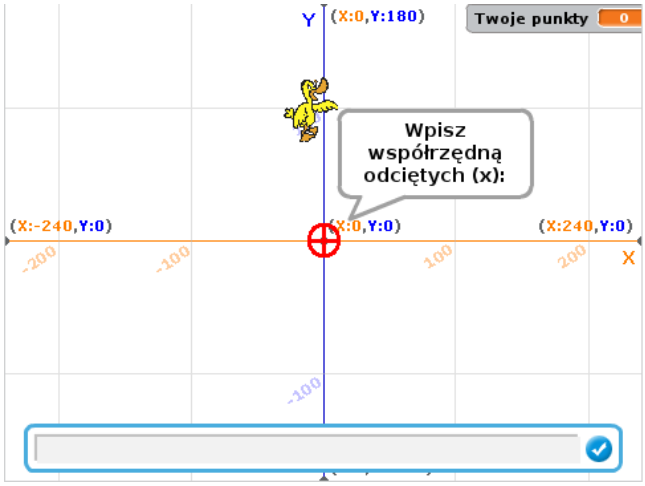


W grze uczestniczą dwie osoby. Wpierw muszą oni wybrać, kto z nich ma zacząć, a następnie na przemian dodają do liczby w puli 1 lub dwa (na początku pula wynosi oczywiście 0). Wygra ten, kto pierwszy powie 20.

Aplikacja stworzona w Scratchu służy tu jako wirtualny przeciwnik w grze, który zachowuje się jak zawodnik nieznający odpowiedniej strategii. Zadaniem użytkownika jest wymyślenie takiej strategii, by zawsze wygrać z komputerem (strategia ta powinna działać bez względu na to, kto będzie zaczynał).

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/programowanie/gra-w-20

Polowanie na kaczki

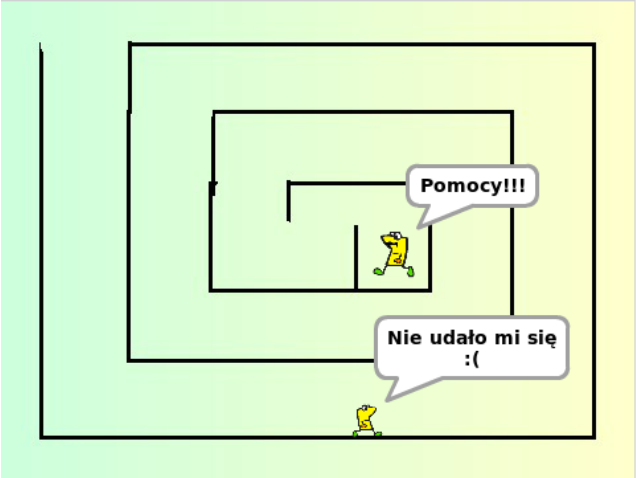


Gra matematyczna polegająca na odczytaniu położenia kaczki i wpisaniu współrzędnych tak, by czerwony celownik najechał na nią. Gra toczy się, aż zawodnik zdobędzie 10 punktów.

Za każde trafienie otrzymuje się jeden punkt, a za pudło tyle samo traci. Jeśli zawodnik dokładnie odczyta współrzędne kaczki, to dostaje od razu 10 punktów.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/programowanie/polowanie-na-kaczki

Labirynt



Typowa gra zręcznościowa, polegająca na przeprowadzeniu jednego duszka przez labirynt tak, by dostał się do swojego kolegi. Każdy może tę grę nieco ulepszyć, zmieniając kształt labiryntu, gdyż jest to zwyczajny duszek. Polecam stosowanie gry na początkowych zajęciach w klasie pierwszej szkoły podstawowej, gdyż dobrze uczy ona klikania lewym klawiszem myszki i stabilnego jej prowadzenia po podłożu.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/programowanie/labirynt

Tabliczka mnożenia dla Ani



Oryginalnie projekt był wykonany dla uczennicy klasy pierwszej szkoły podstawowej, która bardzo usilnie prosiła mnie o przepytanie jej z tabliczki mnożenia do 30, gdyż przygotowywała się do konkursu.

Program ten uzupełniony o proste statystyki, zliczające ile było wszystkich prób odpowiedzi oraz ile z nich było poprawnych może stanowić ciekawy pomysł na uruchomienie rywalizacji między uczniami, w której będą ćwiczyli szybkie mnożenie w pamięci. Oczywiście program bardzo prosto można zremiksować do wersji, gdzie tabliczka mnożenia będzie do 100, 200, 1000 itp.

Zobacz działanie programu:
enauczanie.com/metodyka/scratch/programowanie/tabliczka-mnozenia-dla-ani

Zbiór zadań

W tym dziale zostaną zaprezentowane przykładowe projekty, jakie można zadać uczniom do wykonania w Scratchu. Tematyka zadań oscyluje nie tylko wobec zagadnień informatycznych. Zamieszczone projekty zostały dobrane tak, by pokrywały się z zainteresowaniami uczniów oraz dbały o rozwój ich kreatywnego myślenia.

Grupa 0

Zadanie 1. Przedstaw się
Opowiedz parę informacji o sobie (imię, wiek, hobby, ulubioną potrawę), korzystając ze Scrtcha. Dowolnie zmień swojego duszka i tło sceny. Do stworzenia programu użyj fioletowych bloczków **Powiedz ... przez ... s**.

Zadanie 2. Haiku
Haiku to wiersz składający się z siedemnastu sylab zapisanych w trzech linijkach tekstu (pięć w pierwszej, siedem w drugiej i pięć w trzeciej). Zaanimuj haiku w Scratchu, za pomocą trzech bloczków **Powiedz ... przez ... s**. Pamiętaj, że możesz stworzyć różne postacie i scenerie.

Zadanie 3. Tworzymy scenkę
Za pomocą dwóch duszków i klocków **Powiedz ... przez ... s** i **Czekaj ... s** zaaranżuj dialog między dwoma postaciami. Tło sceny i wygląd swoich duszków dopasuj do tematyki rozmowy. Rozmowa powinna składać się co najmniej z dziesięciu dymków dialogowch.

Zadanie 4. Reklama Scratcha
Jak mogłaby wyglądać reklama Scratcha? Jak przekonałbyś kogoś do jego wypróbowania?

Zadanie 5. Podróże małe i duże
Gdzie byłeś? Gdzie chciałbyś pojechać? Jakie projekty możesz stworzyć w Scratchu o miejscach wartych odwiedzenia?

Zadanie 6. Obrazowanie słów
Jakie projekty w Scratchu możesz stworzyć, by pomóc innym zrozumieć takie słowa jak: *pętla* (w sensie powtarzanej wielokrotnie czynności), *inny*, *trudny*, *czekaj*, *stój*, *obcy*, *piękny*, *przyjaciel*, *deszcz*, *otwarty*, *miłość*, *sztuka*, *syrena*, *instrukcje*, *łatwe*, *moc*, *czas*, *bohater*.

Zadanie 7. Kartka świąteczna
Zaprojektuj interaktywną kartkę świąteczną. Stwórz w tym celu własnego duszka i tło. Nie zapomnij o dodaniu dźwięków.

Zadanie 8. Dziękuję
Czy ktoś ostatnio zrobił dla Ciebie coś dobrego? Czy jesteś komuś za coś wdzięczny? Jak mógłbyś im powiedzieć "dziękuję" za pomocą projektu stworzonego w Scratchu?

Zadanie 9. Pohałasujmy!
Stwórz w Scratchu projekt, zawierający interesujące, nietypowe i inspirujące dźwięki lub muzykę. Możesz w tym celu wykorzystać bloczki przedstawione na rysunku poniżej.



Zadanie 10. Prostokąty
Nie ma tu kół i okręgów, nie ma linii, nie ma też importowanych obrazków. Stwórz projekt, wykorzystując jedynie własnoręcznie narysowane prostokąty (tło sceny powinno być również wykonane z prostokątów).

Zadanie 11. Za dużo, za mało
Stwórz program, w którym użytkownik będzie zgadywał losowo wybraną liczbę przez komputer. Będziesz potrzebował duszka, który powie użytkownikowi, że pomyślał sobie o pewnej liczbie od 1 do 100, a następnie poprosi go o odgadnięcie tej liczby. Duszek powinien mówić “Za dużo” i “Za mało” w momencie, gdy użytkownik będzie źle zgadywał. Gdy użytkownik odgadnie wylosowaną liczbę, duszek powinien zmienić swój kostium. Skorzystaj z bloczków widocznych na rysunku poniżej.



Zadanie 12. 12 wyzwań

Poniżej znajduje się lista dwunastu wyzwań, każda grupa losuje dla siebie jedno z nich, a następnie stara się je jak najlepiej wykonać.

1. Gdy naciśniesz klawisz B – duszek się nieco zwiększy, a gdy naciśniesz S – zmniejszy.
2. Gdy duszek usłyszy głośny dźwięk, powinien zmieniać swój kolor.
3. Gdy duszek jest w górnej ćwiartce sceny, ma mówić “Podoba mi się tu na górze”
4. Gdy duszek dotyka czegoś niebieskiego, zaczyna grać wysoki dźwięk, a gdy dotyka czegoś czerwonego – niski dźwięk.
5. Gdy dwa duszki się zderzają, jeden z nich ma mówić “Przepraszam”.
6. Zawsze gdy kot i pies się do siebie zbliżają, kot ucieka od psa.
7. Gdy klikniesz na tło sceny, pojawi się kwiatek w tym miejscu.
8. Gdy klikniesz na duszku, drugi duszek zacznie tańczyć.
9. Duszek będzie spadał, jakby działała na niego grawitacja, ale zatrzyma się w momencie kontaktu z zielonym podłożem.
10. Duszek podąża za kursorem myszki, ale nigdy się do niego bardzo nie zbliża.
11. Duszek podąża za czerwoną linią.
12. Gdy wynik osiągnie 10, tło sceny się zmieni.

Zadanie 13. Kwadrat i kółko

Jaki program jesteś w stanie stworzyć mając do dyspozycji dwa duszki będące pomarańczowym kwadratem i fioletowym kółkiem. Możesz korzystać tylko z poniższych klocków.



Grupa A

Zadanie 1. Podróżnik

Wykonaj w Scratchu animację opisującą pięć różnych miejsc na świecie. Ustaw tło na mapę świata i spraw, by duszek przemieszczając się między wybranymi przez Ciebie miejscami powiedział dwa zdania o każdym z nich.

Zadanie 2. Wynalazki

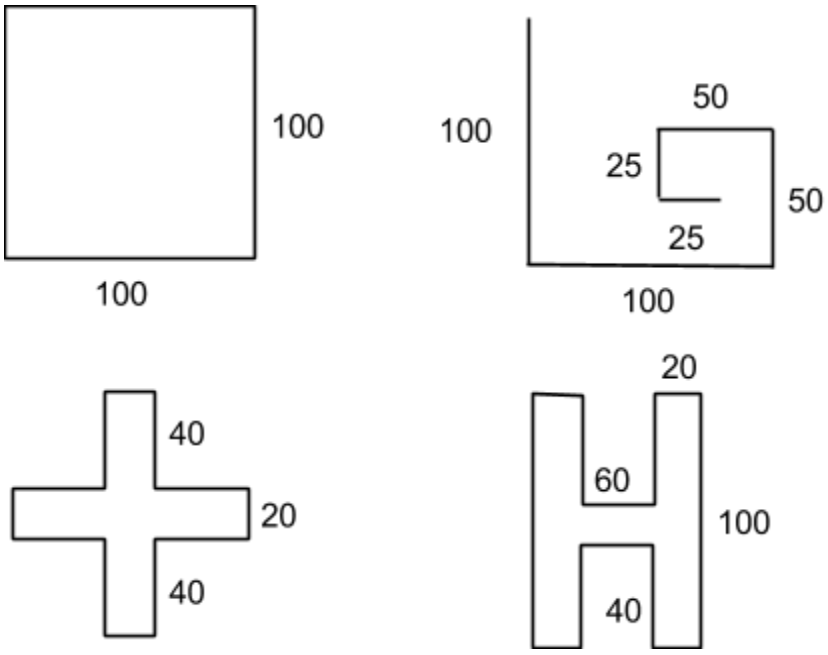
Stwórz program, który będzie prezentował trzy - twoim zdaniem - najważniejsze wynalazki, w historii. Zaprezentuj je odpowiednio, używając możliwości, jakie daje Scratch. Użyj dźwięku, animacji duszka i innych obiektów, zadbaj o odpowiedni wygląd sceny.

Zadanie 3. Pensja

Stwórz program, który będzie wyliczał ile złotych zarobi użytkownik po przepracowaniu danej przez użytkownika liczby godzin. Przyjmij stawkę 13 zł/h. Przykładowo, jeżeli użytkownik poda, że przepracował 10 godzin, to program powinien powiedzieć mu, że zarobił 130 zł.

Zadanie 4. Rysujemy kształty

Narysuj poniższe kształty. Liczby podane obok odcinków to kroki, jakie duszek musi pokonać. Do poruszania duszkiem używaj tylko z klocków **przesuń o … kroków** oraz **ustaw kierunek na**.



Zadanie 5. Piszemy swoje imię
Napisz w Scratchu swoje imię. Do poruszania się używaj klocka **idź do x: … y: …** . Możesz wpierw napisać swoje imię na brudno na arkuszu, po czym odczytasz z niego współrzędne, po jakich ma się poruszać duszek.
Adres arkusza: enauczanie.com/metodyka/scratch/materialy/uklad-wspolrzednych

Zadanie 6. Sztuka z 12 klocków
Stwórz rysunek, używając tylko tych 12 klocków, które widzisz poniżej. Nie musisz używać wszystkich z nich oraz możesz dowolnie zmieniać ich wartości.



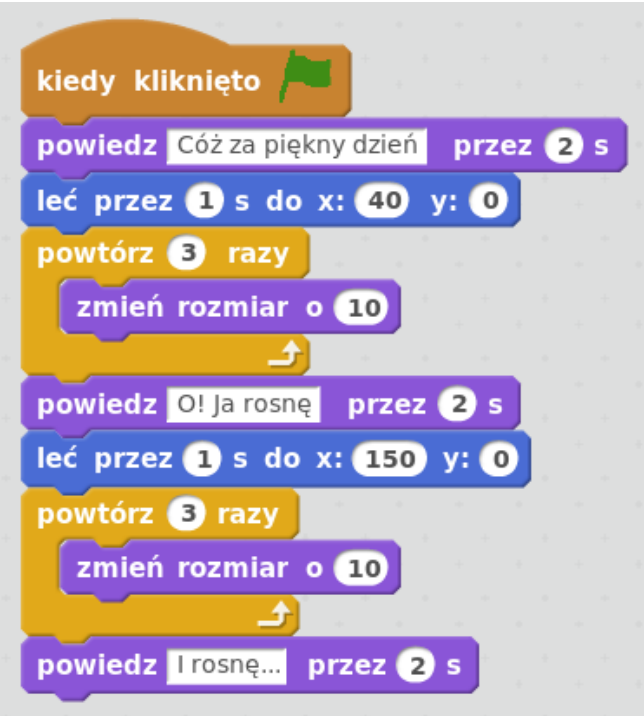
Zadanie 7. Uciekający duszek
Stwórz program, w którym duszek będzie uciekał przed wskaźnikiem myszki, gdy ta będzie się do niego zbliżała na mniej niż 10 kroków.

Zadanie 8. Co jest nie tak z programem?: Arek
Arek chciałby, żeby jego kot obracał się dookoła, gdy naciśnie spację, ale kot się nie porusza. Co się dzieje? Dokonaj odpowiednich zmian w kodzie, by program działał.



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/materialy/co-jest-nie-tak-z-programem

Zadanie 9. Co jest nie tak z tym programem?: Stefcia
Stefcia chce, żeby kot startując od środka planszy, a potem poruszał się w prawo i rósł. Program działa po pierwszym kliknięciu zielonej flagi, ale potem już nie. Co się dzieje? Dokonaj odpowiednich zmian w kodzie, by program działał.



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/materialy/co-jest-nie-tak-z-programem

Zadanie 10. Co jest nie tak z tym programem?: Michał
Michał chce, żeby jego kot tańczył w rytm muzyki, ale kot zaczyna tańczyć dopiero, gdy piosenka się skończy. Gdzie leży problem? Dokonaj odpowiednich zmian w kodzie, by program działał.



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/materialy/co-jest-nie-tak-z-programem

Zadanie 11. Co jest nie tak z tym programem?: Karolina
Karolina chce poruszać kotem: po naciśnięciu strzałki w lewo ma iść w lewo, a strzałki w prawo - w prawo. Chce też, żeby kot mówił, po której stronie planszy się znajduje. Kot się porusza, ale nie podaje poprawnie swojej pozycji. Co się dzieje? Dokonaj odpowiednich zmian w kodzie, by program działał.



Zobacz działanie programu:
enauczanie.com/metodyka/scratch/materialy/co-jest-nie-tak-z-programem

Zadanie 12. Co jest nie tak z tym programem?: Tadek

Tadek chciałby, żeby jego kot najpierw podskakiwał w górę i w dół, a potem poruszał się wzdłuż ekranu, ale kot skacze do przodu. Co się dzieje? Dokonaj odpowiednich zmian w kodzie, by program działał.



Zobacz działanie programu:

enauczanie.com/metodyka/scratch/materialy/co-jest-nie-tak-z-programem

Zadanie 13. 8 bloczków

Jakie programy możesz stworzyć z poniższych 8 bloczków



Zadanie 14. Połącz bloczki, by program działał 1
Połącz wszystkie bloczki tak, żeby kot zmalał, powiedział, że jest mały, a następnie powrócił do swojego normalnego rozmiaru.



Zadanie 15. Połącz bloczki, by program działał 2
Połącz wszystkie bloczki tak, żeby kot spytał się ile jest 6*8, a potem powiedział, czy odpowiedź użytkownika jest dobra.



Zadanie 16. Połącz bloczki, by program działał 3
Połącz wszystkie bloczki tak, żeby kot cały czas wymawiał współrzędne wskaźnika myszki.



na powyższym obrazku, w jednym z zielonych bloczków znajduje się kropka - to nie błąd

Zadanie 17. Połącz bloczki, by program działał 4

Połącz wszystkie bloczki tak, żeby kot rysował czerwony kwadrat.



Grupa B

Zadanie 1. Średnia arytmetyczna
Stwórz program obliczający średnią arytmetyczną z dwóch liczb podawanych przez użytkownika.

Zadanie 2. Kalkulator
Stwórz prosty kalkulator, który będzie obliczał sumę, różnicę, iloczyn i iloraz dwóch liczb podanych przez użytkownika.

Zadanie 3. Liczby parzyste i nieparzyste
Stwórz program sprawdzający, czy liczba podana przez użytkownika jest liczbą parzystą, czy nieparzystą.

Zadanie 4. Wartość bezwzględna
Napisz program obliczający wartość bezwzględną z liczby podanej przez użytkownika.

Zadanie 5. Prostokątność trójkąta
Stwórz program sprawdzający, czy trójkąt o bokach podanych przez użytkownika jest trójkątem prostokątnym.

Zadanie 6. Totolotek
Wykonaj w Scratchu program losujący sześć liczb z przedziału od 1 do 49. Spraw, by duszek wypowiadał je po kolei. Pamiętaj, że nie mogą się one powtarzać.

Zadanie 7. Litery
Napisz program, który poinformuje, czy podany przez użytkownika znak to samogłoska, spółgłoska, czy też coś innego.

Zadanie 8. Silnia
Stwórz program obliczający silnię liczby podanej przez użytkownika.

Zadanie 9. Dodatnia, czy ujemna
Napisz program, który będzie sprawdzał, czy podana przez użytkownika liczba jest dodatnia, ujemna, czy równa 0.

Zadanie 10. Kolejne potęgi
Stwórz program, który po podaniu przez użytkownika liczby naturalnej, będzie wyświetlał jej kolejne potęgi, dopóki użytkownik nie naciśnie klawisza **K**.

Zadanie 11. Dzielniki
Stwórz program wyznaczający wszystkie dzielniki podanej przez użytkownika liczby naturalnej.

Zadanie 12. Liczba cyfr
Stwórz program, który dla podanego przez użytkownika numeru określi liczbę jej cyfr.

Zadanie 13. Palindrom
Użytkownik podaje dowolną liczbę trzycyfrową. Zadaniem programu będzie sprawdzenie, czy ta liczba jest palindromem.

Zadanie 14. Wzór Herona
Utwórz program wyliczający pole trójkąta na podstawie długości jego trzech boków *a*, *b* i *c*, korzystając ze wzoru Herona:

$$S = \sqrt{p(p-a)(p-b)(p-c)}\ ,$$

gdzie *p* oznacza połowę obwodu tego trójkąta. Gdy tego dokonasz, spraw by program przed obliczeniem pola figury sprawdzał, czy z trzech podanych przez użytkownika boków da się zbudować trójkat i, jeśli tak nie będzie - prosił użytkownika o ponowne podanie długości boków.

Zadanie 15. Wirtualny policjant
Stwórz program, który będzie wirtualnym policjantem, rozdającym mandaty. Ma on działać w ten sposób, że użytkownik, po podaniu prędkości z jaką przekroczył tę dozwoloną, dostanie informację o tym jaki wysoki mandat musi zapłacić i ile punktów karnych dostanie. Jako pomocy użyj poniższej tabeli.

Przekroczenie prędkości	Wysokość mandatu	Punkty karne
do 10 km/h	50 zł	0
11 - 20 km/h	100 zł	2
21 - 30 km/h	200 zł	4
31 - 40 km/h	300 zł	6
41 - 50 km/h	400 zł	8
powyżej 50 km/h	500 zł	10

Zadanie 16. Rok przestępny
Napisz program, który po pobraniu roku podanego przez użytkownika stwierdzi, czy jest to rok przestępny, czy nie.

Zadanie 17. Tynk bawełniany
Jedno opakowanie tynku bawełnianego starcza na pokrycie $5m^2$ powierzchni ściany. Stwórz program, który w zależności od podanej przez użytkownika powierzchni ściany, wskaże ile opakowań tynku bawełnianego będzie potrzebne do jej pokrycia. Pamiętaj, że odpowiedź ma być w postaci liczby naturalnej (nikt nie kupi 2,5 opakowania).

Zadanie 18. Włosy

Ludzkie włosy rosną około 0,35 mm dziennie. Stwórz program, który po wpisaniu przez użytkownika żądanej przez niego długości włosów zwróci informację, jak długo musi one te włosy zapuszczać.Zadanie 18. Włosy

Grupa C

Zadanie 1. Dec na Bin
Stwórz program zamieniający dowolną liczbę naturalną na jej odpowiednik zapisany w systemie binarnym.

Zadanie 2. Bin na Dec
Stwórz program zamieniający dowolną liczbę zapisaną w systemie binarnym na jej odpowiednik zapisany w systemie dziesiętnym.

Zadanie 3. Pierwiastki równania kwadratowego
Napisz program zwracający pierwiastki równania kwadratowego podanego w postaci ogólnej.

Zadanie 4. Ciąg Fibonacciego
Stwórz program wyświetlający kolejne liczby ciągo Fibonacciego.

Zadanie 5. Liczba pierwsza
Utwórz program sprawdzający, czy podana przez użytkownika liczba jest liczbą pierwszą.

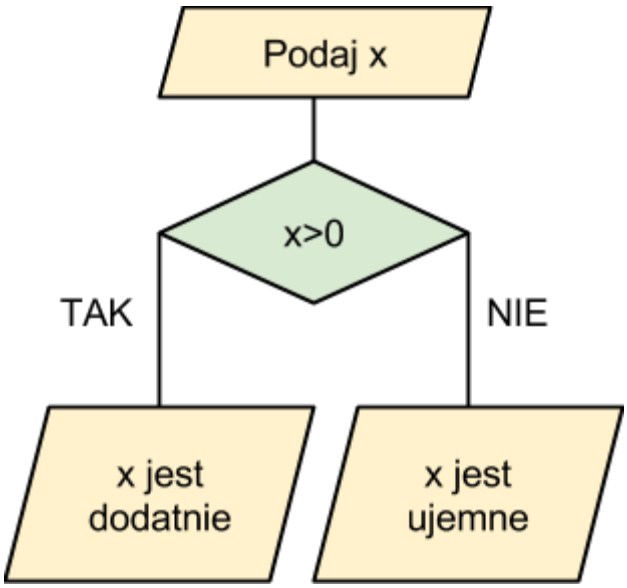
Zadanie 6. Liczby pierwsze
Utwórz program wypisujący *n* początkowych liczb pierwszych, gdzie *n* jest liczbą podaną przez użytkownika.

Zadanie 7. NWD
Stwórz program obliczający NWD dwóch liczb podanych przez użytkownika.

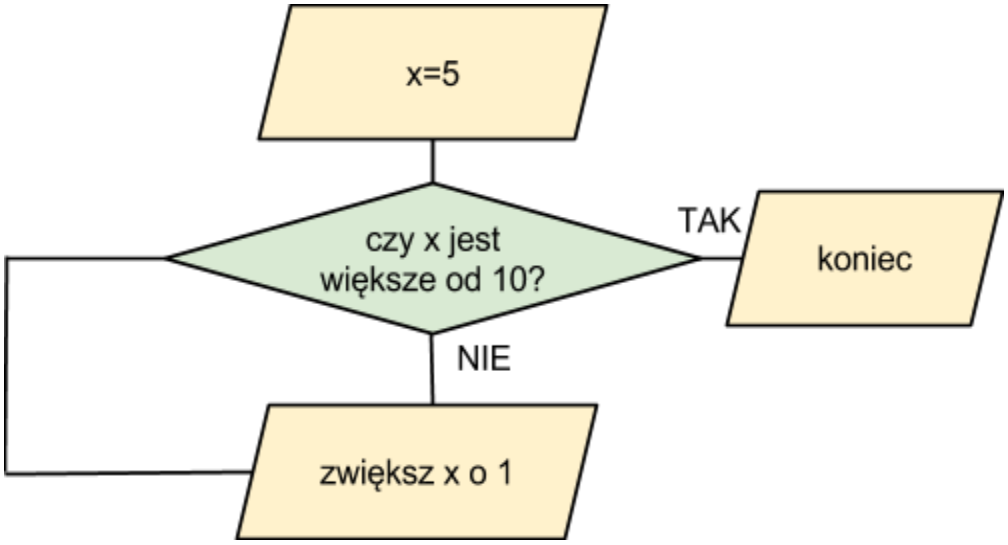
Zadanie 8. Odległosć między punktami
Napisz skrypt, który będzie obliczał odległość między dwoma punktami w kartezjańskim układzie współrzędnych. Punkty mają być zadane przez użytkownika.

Objaśnienia

- **Algorytm** - ciąg jasno zdefiniowanych czynności prowadzących do rządanego rezultatu.
- **Iteracja** - zob. pętla
- **Instrukcja warunkowa** - pozwala na wykonanie pewnej czynności programu w zależności od tego, czy podane w niej twierdzenie jest prawdziwe



- **Pętla** - pozwala na cykliczne powtarzanie pewnej czynności. Może się tak dziać przez określoną liczbę powtórzeń (*powtórz 10 razy*) lub do spełnienia warunku (*powtarzaj aż*)



- **Rekurencja** - W programowaniu nazywamy tak odwoływanie się funkcji do samej siebie. Możemy zobaczyć to na przykład w wyznaczaniu kolejnych wyrazów ciągu Fibonacciego. Definiujemy tam pierwszy i drugi wyraz jako 1, a każdy kolejny wyraz ma być sumą swoich poprzedników.

W ten sposób czwarty wyraz to suma trzeciego i drugiego (czyli 1). Trzeci z kolei to suma pierwszego (1) i drugiego (1). Stąd czwarty wyraz to 1+1+1=4.

O autorze



Nazywam się Piotr Szlagor i jestem nauczycielem matematyki oraz przedmiotów informatycznych. Na co dzień mieszkam w Bielsku-Białej w malowniczej scenerii Beskidów.

Oprócz oczywistych zainteresowań związanych z matematyką i informatyką, pasjonuję się również fotografią. Lubię także uprawiać sport, a w szczególności grać w koszykówkę. Wszystkie zainteresowania staram się łączyć z nowoczesnymi technologiami i dobrodziejstwami, które niesie ze sobą Web 2.0.

Jestem jednym z redaktorów serwisu eNauczanie.com, który poświęcony jest nowoczesnemu nauczaniu i zrzeszają specjalistów z tej dziedziny. Zachęcam do ich odwiedzenia, można tam znaleźć wiele ciekawych materiałów, nawet dla osób nie zajmujących się edukacją.

Specjalizuję się również w tworzeniu i administracji serwisami opartymi o różne platformy (np. Blogger, WordPress, Joomla), obsłudze różnych systemów operacyjnych (Windows, Linux), grafice 2D (Photoshop, Corel Draw), animacji (Google Web Designer) i 3D (Blender, Sketchup) oraz biegłym poruszanie się po świecie nowych technologii.

Przydatne linki

- enauczanie.com/narzedzia/scratch - największy serwis w Polsce poświęcony Scratchowi, a także repozytorium materiałów tego podręcznika.
- scratched.media.mit.edu - anglojęzyczny serwis dla nauczycieli pracujących ze Scratchem. Mnóstwo materiałów, przykładów i dobrych praktyk do wykorzystania na zajęcia.
- info.scratch.mit.edu/pl/Support/Scratch_Cards - karty wycinanki Scratcha. Przód pokazuje zadanie, a tył metodę jego wykonania
- scratched.gse.harvard.edu/guide/download.html - zeszyt ćwiczeń, uczących programowania w Scratchu przygotowany przez Harvard. Jest w języku angielskim, ale ćwiczenia są proste do zrozumienia nawet przez osoby nieanglojęzyczne.