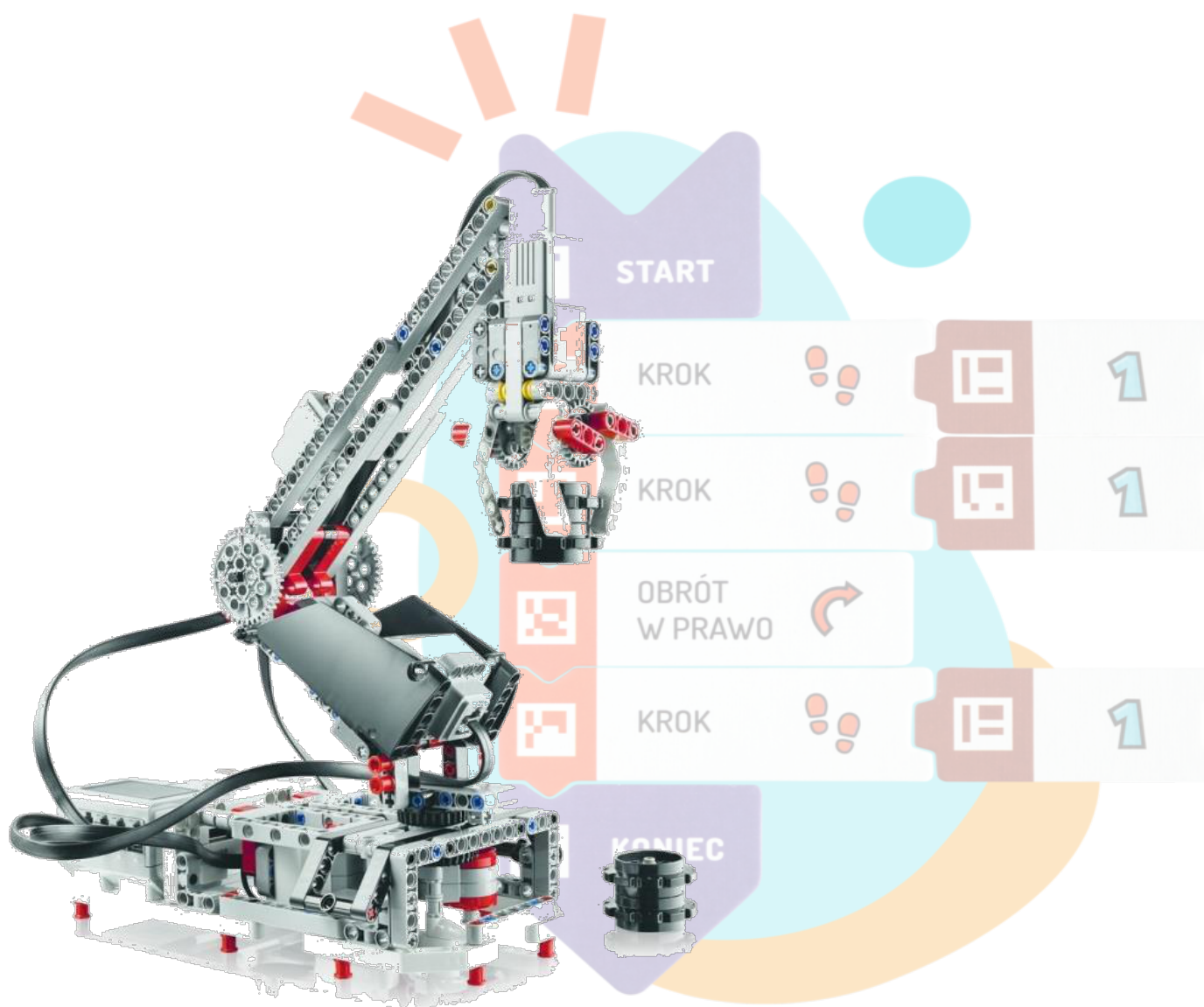


Scenariusze zajęć z uczniami

Programowanie w różnych językach

poradnik dla pracownika gminnej samorządowej instytucji kultury



Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Rozwoju Regionalnego, Programu Operacyjnego Polska Cyfrowa na lata 2014-2020.



Fundusze Europejskie
Polska Cyfrowa



Rzeczpospolita
Polska

Unia Europejska
Europejski Fundusz
Rozwoju Regionalnego



spis treści

WPROWADZENIE

Scenariusz 1	Zabawy z algebrą Boole'a - bramki i operacje logiczne	4-5
Scenariusz 2	Zabawa i nauka programowania ze Scottie Go! - wprowadzenie i organizacja pracy z grą Scottie Go! definicja instrukcji	6-7
Scenariusz 3	Zabawa i nauka programowania ze Scottie Go! - sterowanie bohaterem, tworzenie pierwszych instrukcji, tworzenie liczb, skręcanie, poruszanie do tyłu (moduł I aplikacji)	8-11
Scenariusz 4	Zabawa i nauka programowania ze Scottie Go! - podnoszenie obiektów, skakanie, pętle (moduły: II, III aplikacji)	12-13
Scenariusz 5	Duszek i jego możliwości. Pierwsze kroki w języku Scratch	14-15
Scenariusz 6	Scratch - usystematyzowanie wiedzy, ćwiczenia praktyczne	16-17
Scenariusz 7	Scratch - wielokrotne przetwarzanie, warunki i zdarzenia cz. I	18-19
Scenariusz 8	Scratch - wielokrotne przetwarzanie, warunki i zdarzenia cz. II	20-21
Scenariusz 9	Projektujemy i programujemy grę komputerową dla dwóch graczy	22-24
Scenariusz 10	Budowa dźwigu LEGO Mindstorms	26-27
Scenariusz 11	Programowanie modelu dźwigu	28-19
Scenariusz 12	LEGO Mindstorms - budujemy robota gąsiennicowego	30-31
Scenariusz 13	Na czym polega programowanie tekstowe w Python'ie	32-35
Scenariusz 14	Instrukcje warunkowe, operatory logiczne i operatory porównania w języku Python	36-37
Scenariusz 15	Programowanie i testowanie algorytmów w języku Python	38-40
NOTATKI		41-43

scenariusz 1

czas trwania: 2 godz.

Zabawy z algebrą Boole'a - bramki i operacje logiczne.

Cel ogólny:

Nabywanie wiedzy i umiejętności w zakresie praktycznego zastosowania wartości logicznych w programowaniu pracy komputera.

Cele szczegółowe. Uczeń:

- zna powiązanie pomiędzy budową komputera a logiką matematyczną,
- zna definicje bramki logicznej,
- wykorzystuje tabele prawdy w tworzeniu bramek logicznych,
- opisuje działanie podstawowych bramek logicznych (*AND, NAND, NOT, OR, NOR*),
- rozumie istotę podstawowych funkcji logicznych - alternatywa, koniunkcja, negacja,
- analizuje przebieg sygnału wejściowego w układach prostych i złożonych,
- wyszukuje różne drogi rozwiązania sytuacji problemowej,
- współpracuje w grupie zgodnie z przyjętymi zasadami.

Rola prowadzącego:

- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- poszukiwanie alternatywnych rozwiązań problemu.

Rola uczniów (w wieku 10-18 lat):

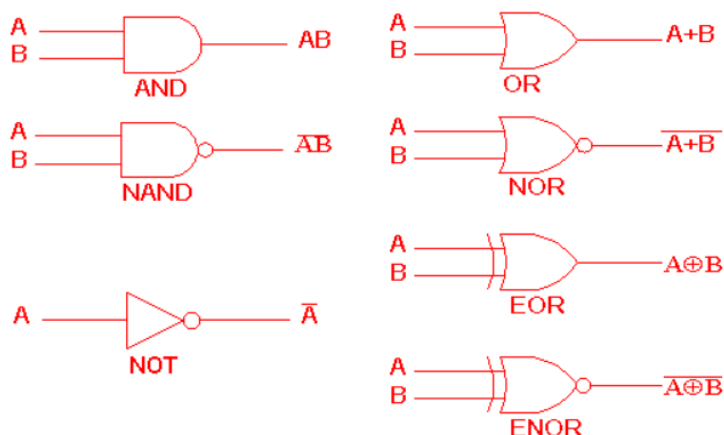
- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- poszukiwanie alternatywnych rozwiązań problemu.

Przygotowanie do zajęć – środki dydaktyczne:

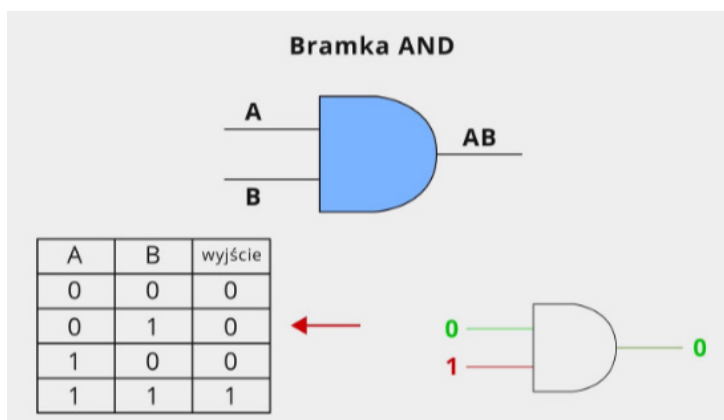
- komputery/laptopy z oprogramowaniem i dostępem do Internetu,
- projektor,
- prezentacja multimedialna,
- pytania do quiz'u.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.
2. Przedstawienie tematu i celów zajęć.
3. Prezentacja multimedialna – graficzna demonstracja podstawowych bramek logicznych.
Omówienie zasady ich konstrukcji i działania.



4. Film instruktażowy: „Bramki logiczne, czyli nowe znaczenie zera i jedynki”.



Źródło: kadr z materiału video: www.epodreczniki.pl; 27.07.2012

5. **Ćwiczenie nr 1** - quiz – prowadzący dzieli uczniów na grupy, uczniowie z każdej grupy losują kartki z pytaniami, następnie odpowiadają na nie na podstawie obejrzanego filmu.

Przykładowe pytania:

- Jakie są podstawowe bramki logiczne?
- Czy określenie „bramka dwustanowa” oznacza, że ma ona dwa wejścia?
- Czy istnieje bramka, która ma tylko jedno wejście? Jeśli tak, podaj jaką.
- Czy bramki mogą mieć więcej niż dwa wejścia?
- Czy bramki mogą mieć więcej niż jedno wyjście?
- Czy bramka (AND) realizuje funkcję sumy logicznej?
- Co to jest sygnał wejściowy?
- Czy stan 1 oznaczamy literą „W” od wysoki?
- Czy do opisu działania bramki używamy logiki matematycznej?

Na koniec każda grupa liczy, ile zdobyła punktów.

Prezentacja odpowiedzi na forum grupy.

Wspólna dyskusja na temat poprawności udzielanych odpowiedzi.

6. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji dotyczących bramek i operacji logicznych,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 2

czas trwania: 2 godz.

Zabawa i nauka programowania ze *Scottie Go!*

– wprowadzenie i organizacja pracy z grą *Scottie Go!* definicja instrukcji.

Cel ogólny:

Wyposażenie uczniów w wiedzę umożliwiającą tworzenie instrukcji w grze edukacyjnej *Scottie Go!*.

Cele szczegółowe. Uczeń:

- zna znaczenie pojęcia instrukcji,
- wie jak zastosować parametr przy tworzeniu instrukcji konsekwencji działań,
- potrafi budować komunikaty z wykorzystaniem elementów instrukcji w *Scottie Go!*,
- podaje przykłady instrukcji związanych z codziennymi czynnościami,
- rozwiązuje problem wyzwania poprzez zbudowanie najprostszej instrukcji,
- przestrzega wyznaczonych zasad współpracy w grupie.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie tabletów z zainstalowaną aplikacją *Scottie Go!* oraz gier *Scottie Go!*,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- wspieranie uczniów w rozwiązywaniu pojawiających się trudności, moderowanie pracy grupy,
- budowanie atmosfery współpracy pomiędzy uczniami.

Rola uczniów (w wieku 10-18 lat):

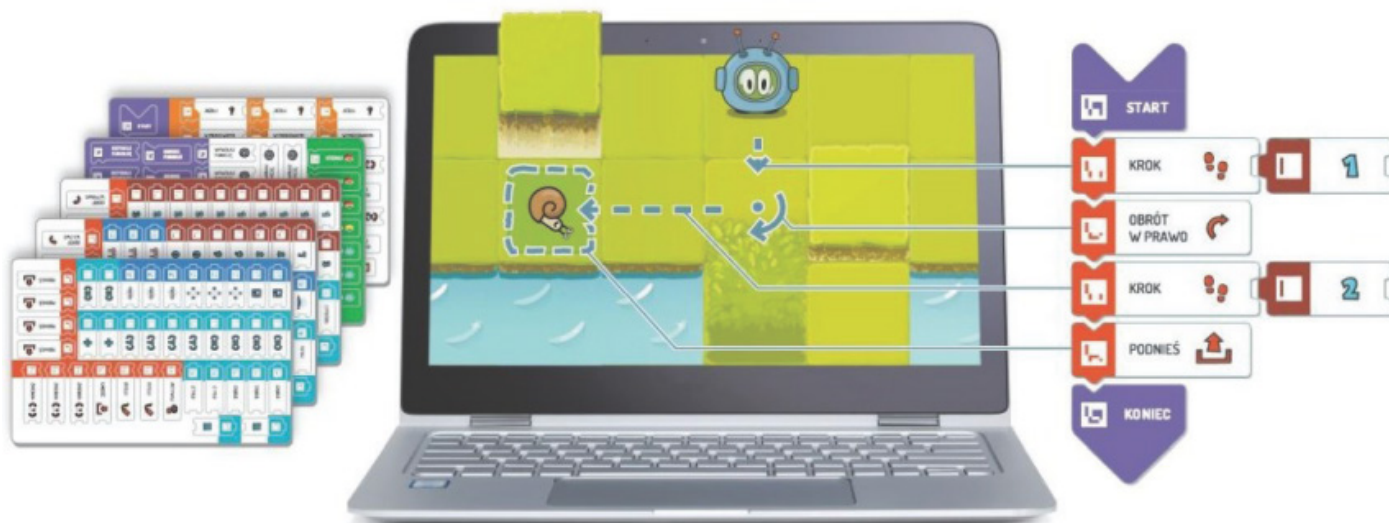
- zapoznanie się z instrukcją instalacji aplikacji
- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach, przestrzeganie zasad bezpiecznego korzystania z technologii IT.

Przygotowanie do zajęć – środki dydaktyczne:

- co najmniej jeden tablet o parametrach, które pozwolą zainstalować aplikację *Scottie Go!*,
- aplikacja mobilna *Scottie Go!*,
- minimum 1 zestaw gry *Scottie Go!*,
- projektor multimedialny.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.
2. Zapoznanie z tematem i celami zajęć.
3. Burza mózgów – dyskusja wokół znaczenia pojęcia instrukcja.
4. **Ćwiczenie nr 1** - zabawa w programowanie unplugged. Wyznaczenie roli robota i określanie przez uczniów instrukcji (komend) czynności mycia zębów.
5. Prowadzący przedstawia grę *Scottie Go!* i opowiada uczniom historię głównego bohatera gry *Scottiego*.



6. Uczniowie pobierają i instalują na tablecie aplikację *Scottie Go!*.
7. Prowadzący informuje uczniów, że po zainstalowaniu aplikacja nie wymaga ona dostępu do Internetu (działa *offline*).

8. Demonstracja klocków przez prowadzącego, służących do tworzenia:

- instrukcji prostych - pojedyncza czynność/operacja do wykonania np.: krok, obrót w lewo, obrót w prawo itd.
- instrukcji złożonych - rozpoczynają się i kończą klockiem w kolorze pomarańczowym, ich przykładem są pętle powtórz itd. lub instrukcja warunkowa.



9. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji w zakresie stosowania instrukcji,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 3

czas trwania: 2 godz.

Zabawa i nauka programowania ze *Scottie Go!* – sterowanie bohaterem, tworzenie pierwszych instrukcji, tworzenie liczb, skręcanie, poruszanie się do tyłu (moduł I aplikacji).

Cel ogólny:

Kształcenie umiejętności wykorzystania zdobytej wiedzy do budowania przez uczniów instrukcji w grze edukacyjnej *Scottie Go!*.

Cele szczegółowe. Uczeń:

- stosuje parametr przy tworzeniu instrukcji konsekwencji działań,
- skanuje aparatem utworzoną uprzednio instrukcję,
- prowadzi obserwację rozwiązania zadania na ekranie tabletu i dostrzega prawidłowość rozwiązane zadania,
- dokonuje weryfikacji utworzonej instrukcji na ekranie i w uzasadnionych przypadkach wprowadza zmiany,
- rozwiązuje problem „Wyzwania” poprzez zbudowanie najprostszej instrukcji,
- przestrzega wyznaczonych zasad współpracy w grupie.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszający, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie tabletów z zainstalowaną aplikacją *Scottie Go!* oraz gier *Scottie Go!*,
- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- wspieranie uczniów w rozwiązywaniu pojawiających się trudności, moderowanie pracy grupy,
- włączenie do aktywnego udziału w zajęciach, utworzenie zespołów do pracy w grupach zadaniowych (zespołach i parach).

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach, przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- poszukiwanie alternatywnych rozwiązań problemu.

Przygotowanie do zajęć – środki dydaktyczne:

- co najmniej jeden tablet o parametrach, które pozwolą zainstalować aplikację *Scottie Go!*,
- aplikacja mobilna *Scottie Go!*,
- minimum 1 zestaw gry *Scottie Go!*,
- projektor multimedialny.

Opis zadań i sposób ich realizacji:

Przebieg zajęć:

1. Praca uczniów w aplikacji *Scottie Go!*
2. Komentarz prowadzącego (do zadania 1-3) stanowiący o potrzebie wykorzystania parametru jako wartości, cyfry/ liczby z jaką dana instrukcja ma działać.
3. Instruktaż prowadzącego w zakresie budowy gry:
 - „Wyzwanie” – na ekranie tabletu pojawia się problem nazwany „Wyzwaniem” – aby wykonać zadanie nasz bohater *Scottie* musi dojść do pola oznaczonego symbolem „X”.



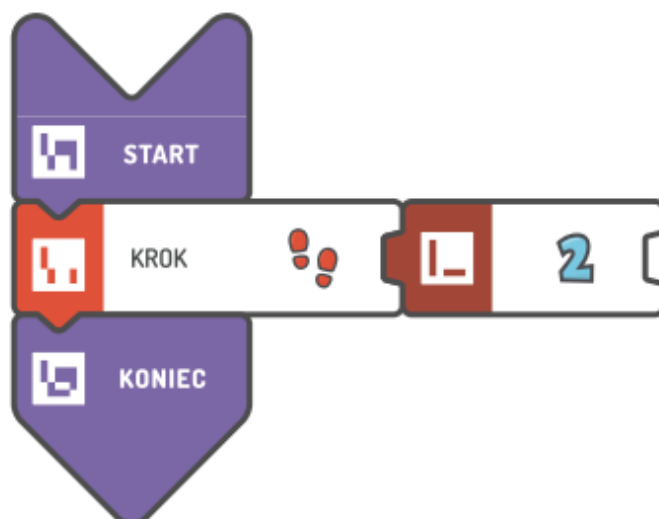
- Każde pole to jeden „krok robota”.



- Z klocków *Scottie Go!* uczniowie układają jak najkrótszą instrukcję, aby rozwiązać „Wyzwanie” i tym samym ukończyć zadanie.
- Instrukcja pisana dla Scottiego powinna rozpoczynać się klockiem START i kończyć klockiem KONIEC.

4. **Ćwiczenie nr 1** - uczniowie układają instrukcję dla Scottiego (w aplikacji zadanie nr 1).

Przykładowa instrukcja:



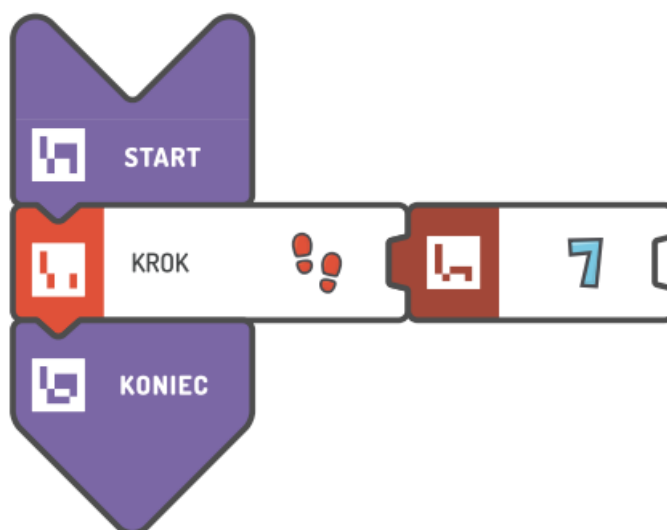
5. Skanowanie instrukcji aparatem. Obserwacja rozwiązania zadania na ekranie tabletu.
Dokonywanie analizy utworzonej instrukcji na ekranie i wprowadzanie zmian w uzasadnionych przypadkach.
Argumentowanie przyjętych rozwiązań na forum grupy.



6. **Ćwiczenie nr 2** - uczniowie układają instrukcje (zadanie 2 modu 1 w aplikacji) przy pomocy klocków, a następnie skanują je tak jak w zadaniu pierwszym.



Przykładowa, poprawna instrukcja:

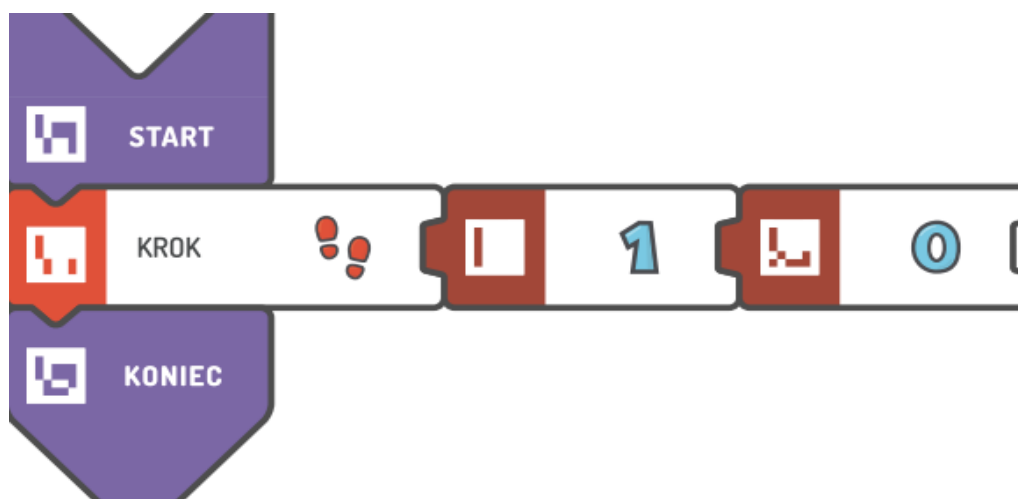


7. Prowadzący prezentuje praktyczne rozwiązania dotyczące, tworzenia parametrów w postaci liczby poprzez łączenie klocków oraz stosowanie dodawania i odejmowania.

8. **Ćwiczenie nr 3** - uczniowie w parach rozwiązują zadanie nr 3 (moduł I) w aplikacji *Scottie Go!*



Przykładowa, poprawna instrukcja:



9. Prowadzący prezentuje praktyczne rozwiązania dotyczące, tworzenia instrukcji w zakresie:

• skręcania:



• poruszania się do tyłu:



10. **Ćwiczenie nr 4** - uczniowie z wykorzystaniem klocków i tabletu układają instrukcję dla wyzwań w zadaniach od 4 do 10 (moduł I) w aplikacji *Scottie Go!*

11. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji m.in. w zakresie procedury budowania parametru,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 4

czas trwania: 2 godz.

Zabawa i nauka programowania ze *Scottie Go!* – podnoszenie obiektów, skakanie, pętle, (moduły: II, III aplikacji).

Cel ogólny:

Nabycie przez uczniów wiedzy i umiejętności kompleksowego zastosowania klocków odpowiadających za podnoszenie obiektów, skakanie, budowanie pętli.

Cele szczegółowe. Uczeń:

- buduje instrukcje z wykorzystaniem nowych elementów w *Scottie Go!*,
- tworzy autorskie instrukcje do rozwiązywania wybranych problemów,
- przedstawia własną koncepcję realizacji „Wyzwania”,
- rozwiązuje problem „Wyzwania” poprzez zbudowanie najprostszej instrukcji,
- przestrzega wyznaczonych zasad współpracy w grupie.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedośłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie tabletek z zainstalowaną aplikacją *Scottie Go!* oraz gier *Scottie Go!*,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- włączenie do aktywnego udziału w zajęciach.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- poszukiwanie alternatywnych rozwiązań problemu.

Przygotowanie do zajęć – środki dydaktyczne:

- co najmniej jeden tablet o parametrach, które pozwolą zainstalować aplikację *Scottie Go!*,
- aplikacja mobilna *Scottie Go!*,
- minimum 1 zestaw gry *Scottie Go!*,
- projektor multimedialny.

Opis zadań i sposób ich realizacji:

1. Przedstawienie uczniom założeń modułu II aplikacji *Scottie Go!*



2. Demonstracja nowych elementów do użycia w Instrukcji.

3. **Ćwiczenie nr 1** - zbudowanie instrukcji dla „Wyzwań” w zadaniach od 1 do 4 (moduł II) w aplikacji *Scottie Go!* z wykorzystaniem klocka SKOCZ.

4. Używając klocka „SKOCZ”, Scottie przemieszcza się z pola 1 do pola 3 odbijając się od klocka nr 2.

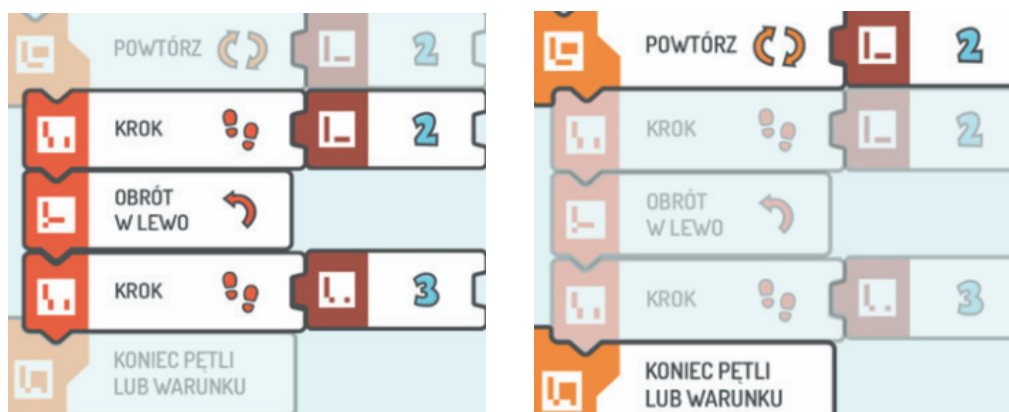


5. Ćwiczenie nr 2 - zbudowanie instrukcji dla wyzwań w zadaniach od 5 do 10 (moduł II) w aplikacji *Scottie Go!*

6. Prowadzący moderuje dyskusję, pyta uczniów o ich pomysły dotyczące możliwych rozwiązań, nowych klocków, które mogłyby pomóc.

Prowadzący demonstruje wykorzystanie pętli POWTÓRZ. Podaje definicję pętli. Aby poprawnie użyć klocków pętli należy wskazać początek i koniec fragmentu programu, który ma zostać powtórzony (za pomocą klocków:

POWTÓRZ i KONIEC PĘTLI LUB WARUNKÓW). Dodatkowo można również wskazać, ile razy fragment programu ma zostać powtórzony (poprzez dodanie do klocka PĘTLA klocka z ilością powtórzeń).



7. **Ćwiczenie nr 3** - uczniowie układają instrukcje dla „Wyzwań” w zadaniach 1 i 2 (moduł III) – w aplikacji *Scottie Go!*

8. **Ćwiczenie nr 4** - uczniowie układają instrukcje i rozwiązują zadanie nr 3 (moduł III) - w aplikacji *Scottie Go!*

Przykładowe rozwiązanie zadania nr 3:



9. Dla uczniów prejawiających zainteresowania omawianą tematyką przedstawione zostaną złożone operacje takie jak: pętle zagnieżdżone, pętle warunkowe, zmienne, działania matematyczne na zmiennych, używanie jednej i wielu instrukcji warunkowych oraz zadań od 4 do 8 (moduł III) oraz modułów od IV do IX.

10. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji m.in. w zakresie procedury budowania instrukcji z elementami „SKOCZ” oraz „POWTÓRZ”;
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 5

czas trwania: 2 godz.

Duszek i jego możliwości. Pierwsze kroki w języku Scratch.

Cel ogólny:

Przekazanie uczniom wiedzy i umiejętności w zakresie podstaw samodzielnego programowania wizualnego (układanie algorytmów w środowisku *Scratch*).

Cele szczegółowe. Uczeń:

- wydaje polecenia „duszkowi” w środowisku *Scratch*,
- poznaje środowisko programowania *Scratch*,
- poznaje elementy interfejsu i ich znaczenie,
- zna określenia „duszek” i „scena”,
- używa podstawowych bloków akcji do poruszania „duszkami” po scenie,
- animuje proste ruchy „duszków”.

Rola prowadzącego:

- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- przygotowanie projektów w środowisku *Scratch*, stanowiących podstawę zilustrowania jego budowy, podstawowych funkcji i możliwości,
- inspirowanie ucznia do pracy w środowisku programowania w języku *Scratch*,
- stwarzanie warunków do modyfikowania przykładowego projektu,
- włączenie do aktywnego udziału w zajęciach,
- utworzenie zespołów do pracy w grupach zadaniowych (zespołach i parach).

Rola uczniów (w wieku 10-18 lat):

- uprzednie zapoznanie się z instrukcją instalacji aplikacji *Scratch*,
- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego problemu,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami.

Przygotowanie do zajęć – środki dydaktyczne:

komputery/laptopy z oprogramowaniem i dostępem do Internetu, zainstalowana aplikacja *Scratch*.

Strona logowania się do programu Scratch: <https://scratch.mit.edu/>

Strona wymaga zalogowania się i podania zarejestrowanego adresu mailowego.



Opis zadań i sposób ich realizacji:

1. Prowadzący przedstawia uczniom możliwości środowiska języka Scratch demonstrując działanie kilku wcześniej wybranych projektów zarówno w wersji offline jak i online.
2. W zależności od poziomu zaawansowania uczniów, część z nich może wykonać wszystkie poniższe zadania, a część może je skomasować w większy projekt. Istnieje możliwość połączenia uczniów w pary, by ci bardziej zaawansowani pomogli początkującym.

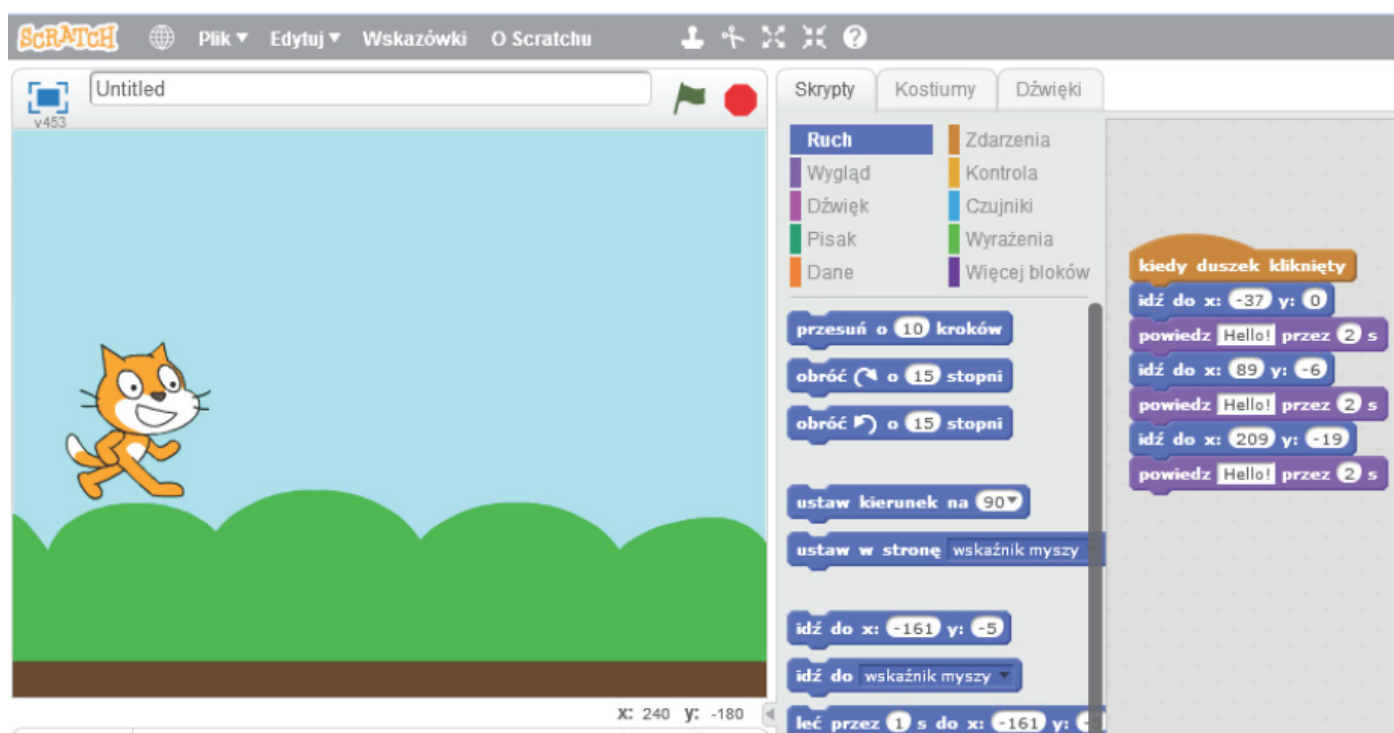
W parach/grupach, mogą oni utworzyć wspólny projekt:

- pierwszy prosty program,
- wybór/zmiana „duszka”,
- wybór/zmiana tła,
- przegląd dostępnych bloków i próby skorzystania z nich - eksperymentowanie.

3. Prowadzący łączy uczniów w pary/grupy – w każdej przynajmniej jest jeden uczeń zaawansowany, który stawiał już pierwsze kroki w programowaniu.

Jeśli nie ma uczniów zaawansowanych, prowadzący wyznacza uczniom prostsze zadanie, tak aby na podstawie znajomości poleceń z łamigłówek Godziny Kodowania, utworzyli program dla „duszka”, który naśladuje którąś z postaci z Godziny Kodowania.

W poniższym projekcie/programie „duszek” skacze po pagórkach i mówi Hello. Zadaniem uczniów zaawansowanych jest wyznaczenie współrzędnych punktów dla „duszka”. Zadaniem uczniów początkujących, jest ułożenie sekwencji zdarzeń dla „duszka” – „duszek” może poruszać się krokami i skręcać, może mówić, dłużej lub krócej, itp.



Po lewej stronie programu znajduje się scena do rozgrywania projektów, która nie ma żadnych ograniczeń, uczeń sam ją tworzy w projekcie. Na środku planszy jest wybór poleceń, po prawej stronie jest obszar roboczy, na którym tworzymy projekt układając sekwencję zdarzeń. Prowadzący powinien dołożyć starań, by uczniowie początkujący nie poczuli się zagubieni w nowej przestrzeni programowania.

4. Podsumowanie i ewaluacja zajęć:

- Jaki projekt udało mi się dziś wykonać?
- Co wydało mi się szczególnie interesujące?
- Jakiego programu chciałam/chciałbym utworzyć?
- Co sprawiło mi problemy?

scenariusz 6

czas trwania: 2 godz.

Scratch – usystematyzowanie wiedzy, ćwiczenia praktyczne.

Cel ogólny:

Usystematyzowanie wiedzy i umiejętności z zakresu języka programowania *Scratch*.

Cele szczegółowe. Uczeń:

- określa parametry ilościowe: wielkość figur, liczbę powtórzeń,
- umie znaleźć powtórzenia – znajduje sekwencje tych samych elementów,
- opisuje figury za pomocą ciągów wizualnych instrukcji z liczbami,
- analizuje obrazki i rozkłada je na sekwencje czynności potrzebnych do ich wykonania zgodnie z założoną regułą,
- tworzy w środowisku programowania *Scratch* proste figury o podanych parametrach, poznaje i stosuje polecenia (przyłóż pisak, podnieś pisak, zmienia kolory pisaka, korzysta z pętli), aby powtórzyć daną sekwencję działań „duszka”,
- prezentuje efekty swojej pracy w środowisku *Scratch* innym uczniom.

Rola prowadzącego:

- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- przygotowanie kart pracy z siatką punktów i linii, plansz w kratkę do zadań realizowanych poza komputerami,
- inspirowanie uczniów do pracy w środowisku programowania w języku *Scratch*,
- stwarzanie warunków do prezentacji przykładowych rozwiązań programowych w środowisku *Scratch*,
- przygotowanie przykładowych rozwiązań dla uczniów o zróżnicowanym poziomie zaawansowania,
- tworzenie grup zadaniowych o różnym poziomie zaawansowania uczniów.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodnie z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianego na forum projektu argumentując zasadność dokonanych wyborów liczby powtórzeń,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery/laptopy z oprogramowaniem i dostępem do Internetu,
- zainstalowana aplikacja *Scratch*,
- karty pracy.

Opis zadań i sposób ich realizacji:

Prowadzący przekazuje uczniom przygotowane przez siebie karty pracy w postaci siatki punktów i linii, planszy w kratkę.

Ćwiczenie nr 1:

do wykonania ćwiczenia potrzebne są siatki punktów i linii, uczniowie tworzą figury z linii jako wynik poleceń, np. 3 w lewo, 2 w prawo. Prowadzący przygotowuje polecenia o różnym stopniu trudności, uczniowie pracują indywidualnie lub w parach, na zakończenie uczniowie prezentują i omawiają swoje prace.

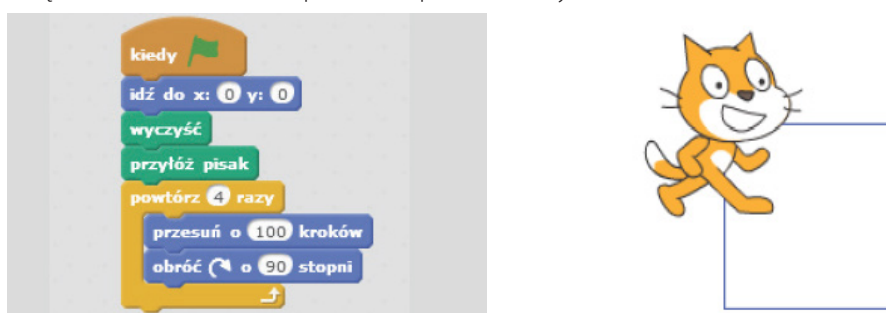
Ćwiczenie nr 2:

do wykonania ćwiczenia potrzebne są plansze w kratkę, posłużą do wyjaśnienia na czym polega oznaczanie punktów na planszy dwoma liczbami (zwanymi współrzędnymi), popularnie oznaczanymi przez x i y – tak jest w środowisku *Scratch*. Zadanie uczniów polega na przesuwaniu „duszka” na określone pozycje wyrażone w współrzędnych x:0 y:0. Prowadzący przygotowuje polecenia o różnym stopniu trudności, uczniowie pracują indywidualnie lub w parach, na zakończenie uczniowie prezentują i omawiają swoje prace.

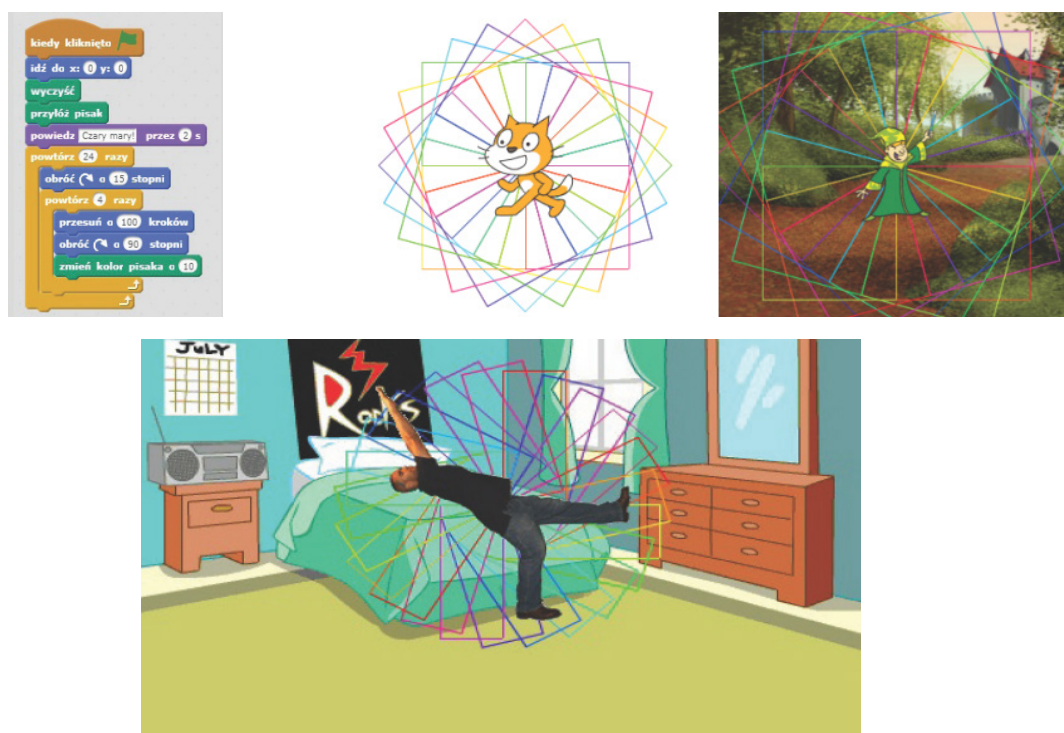
Ćwiczenie nr 3:

uczniowie programują w środowisku *Scratch*: rysowanie figur geometrycznych:

- przykładowe rozwiązanie dla uczniów na poziomie podstawowym:



- przykładowe rozwiązanie dla uczniów na poziomie zaawansowanym (pętla zagnieżdżona):



1. Uczniowie prezentują na forum grupy swoje projekty.

2. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji m.in. w zakresie etapów tworzenia kodu w języku *Scratch*,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 7

czas trwania: 2 godz.

Scratch – wielokrotne przetwarzanie, warunki i zdarzenia. cz. I.

Cel ogólny:

Usystematyzowanie wiedzy oraz podniesienie kompetencji informatycznych poprzez rozwiązywanie skomplikowanych łamigłówek (zdarzenia warunkowe).

Cele szczegółowe. Uczeń:

- opisuje i programuje sytuacje, w których powtarzane są powtarzające się czynności (zagnieżdżane pętle),
- przewiduje alternatywne zakończenia historyjek ze względu na spełniony/niespełniony warunek,
- poznaje warunki w zabawie ruchowej – podejmowanie czynności pod danym warunkiem,
- znajduje informacje w sieci (wyszukiwarka Google, prowadzący wskazuje odpowiednie strony), które posłużą do formułowania warunków,
- układa historyjki i programy z reakcją na zachodzące zdarzenia.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- przygotowanie kart pracy zawierających zadania programistyczne,
- inspirowanie uczniów do pracy w środowisku *Scratch*,
- stworzenie warunków do twórczego rozwiązywania problemów.

Rola uczniów (w wieku 10-18 lat):

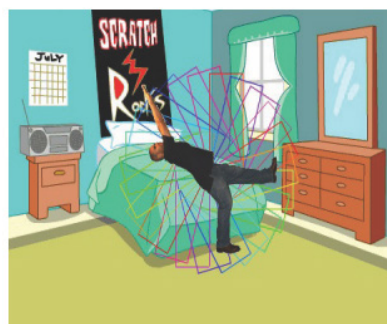
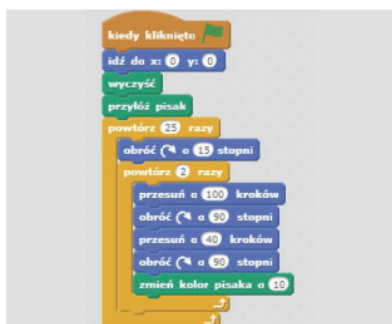
- postępowanie zgodnie z wyznaczonymi etapami pracy,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery/laptopy z oprogramowaniem i dostępem do Internetu,
- zainstalowana aplikacja *Scratch*,
- karty pracy.

Opis zadań i sposób ich realizacji:

1. Prowadzący stwarza sytuację, w której instrukcje powtarzane w jednym bloku mają być również powtarzane wielokrotnie, np. gdy chcemy narysować wiele kwadratów.
2. **Ćwiczenie nr 1:** prowadzący rozdaje uczniom przygotowane przez siebie karty pracy, na których uczniowie mają zaprojektować różne podobne sytuacje, np. wiele takich samych kwadratów.



3. **Ćwiczenie nr 2:** prowadzący prezentuje uczniom stworzoną przez siebie grę w Scratch'u, w której „kot goni mysz”. Następnie uczniowie samodzielnie programują analogiczny projekt.
4. **Ćwiczenie nr 3:** prowadzący prosi uczniów o podane przykłady czynności warunkowych oraz zdarzeń, które mogłyby wystąpić w planowanych przez nich projektach.
5. Podsumowanie i ewaluacja zajęć:
 - zebranie i uporządkowanie istotnych informacji m.in. zastosowania pętli zagnieżdżonych,
 - ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 8

czas trwania: 2 godz.

Scratch – wielokrotne przetwarzanie, warunki i zdarzenia cz. II.

Cel ogólny:

Konstruowanie autorskich projektów z wykorzystaniem funkcjonalności środowiska *Scratch*.

Cele szczegółowe. Uczeń:

- przewiduje alternatywne zakończenia historii ze względu na spełniony/niespełniony warunek,
- znajduje informacje w sieci (np. wyszukiwarka *Google*, prowadzący może wskazać odpowiednie strony), które posłużą do formułowania warunków,
- wykorzystuje możliwości środowiska *Scratch* z uwzględnieniem animacji i interakcji w konstruowanych projektach,
- tworzy w środowisku programowania *Scratch* interakcje „duszków” za pomocą bloków warunkowych „jeżeli”, aktywowanie „duszka” za pomocą kliknięć, dialog „duszków”, zmiana kostiumów i tła pod określonym warunkiem.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedoслyszający, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- przygotowanie kart pracy dla uczniów zawierających historyjki, które wymagają uzupełnienia fragmentów dotyczących dokonywania wyborów i podejmowania decyzji wariantowych, uzależnionych od pewnych warunków,
- utworzenie zespołów do pracy w grupach zadaniowych (zespołach i parach).

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodnie z wyznaczonymi etapami pracy,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery/laptopy z oprogramowaniem i dostępem do Internetu,
- zainstalowana aplikacja *Scratch*,
- karty pracy.

Opis zadań i sposób ich realizacji:

1. Ćwiczenie nr 1:

prowadzący rozdaje przygotowane karty pracy, następnie uczniowie uzupełniają fragmenty, które dotyczą wyborów i podejmowania decyzji wariantowych, uzależnionych od pewnych warunków (zadanie może być wykonane w parach).

2. Ćwiczenie nr 2:

uczniowie (w grupach 2-4 osoby) inspirowani dotychczasowymi opowieściami szkicują sytuacje, które chcieliby zrealizować w postaci projektów w środowisku języka *Scratch*, z uwzględnieniem różnych możliwości tego środowiska, które przejawiają się różnorodnymi aktywnościami „duszka/duszków”.

3. Naśladując idee twórców środowiska *Scratch*, prowadzący prezentuje przykładowe projekty utworzone w tym środowisku, które mogą być inspiracją dla uczniów. Powinny to być prezentacje podstawowych elementów i aktywności w *Scratch'u*.

4. Ćwiczenie nr 3:

uczniowie tworzą projekt w środowisku *Scratch*, w którym:

- występuje wiele postaci „duszków”,
- wykonują one czynności powtarzające się i warunkowe,
- uruchamiane są jako zdarzenia.

Dodatkowe efekty – uczniowie zaawansowani: kolorowe, dźwiękowe, ruchowe, zmiany kostiumów.

Uczniowie pracują w parach: początkujący + zaawansowany.

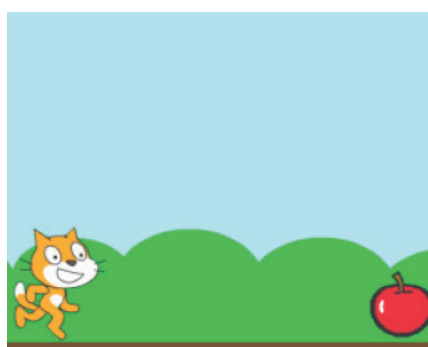
Poniżej przykładowy projekt z wymienionymi powyżej elementami.

Uczeń początkujący uruchamia taki projekt i rozpoznaje, jakie będą potrzebne skrypty do jego wykonania, następnie w parach uczniowie realizują swój projekt, podobny lub zmieniony:

5. Uczniowie prezentują swoje prace w repozytorium środowiska *Scratch*.

6. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji,
- ewaluacja zajęć z wykorzystaniem nowych technologii.



scenariusz 9

czas trwania: 2 godz.

Projektowanie i programowanie gry komputerowej dla dwóch graczy.

Cel ogólny:

Praktyczne zastosowanie zdobytej wiedzy i umiejętności w zakresie wykorzystania aplikacji *Scratch* - tworzenie reguł gry komputerowej w środowisku programistycznym *Scratch*.

Cele szczegółowe. Uczeń:

- znajduje powiązanie pomiędzy budową komputera a logiką matematyczną,
- zna definicje bramki logicznej,
- opisuje działanie podstawowych bramek logicznych,
- zna podstawowe funkcje logiczne takie jak alternatywa, koniunkcja, negacja,
- analizuje przebieg sygnału wejściowego w układach prostych i złożonych.

Rola prowadzącego:

- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- moderowanie konstruktywnej dyskusji wokół poruszanych problemów programistycznych,
- budowanie atmosfery współpracy pomiędzy uczniami.

Rola uczniów (w wieku 10-18 lat):

- przestrzeganie kolejnych etapów pracy wyznaczających schemat postępowania programistycznego,
- włączanie się do dyskusji w zakresie omawianego problemu,
- argumentowanie dokonanych wyborów programistycznych,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- poszukiwanie alternatywnych rozwiązań problemu.

Przygotowanie do zajęć – środki dydaktyczne:

- laptop,
- projektor,
- stanowiska komputerowe z dostępem do Internetu.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.

2. Przedstawienie tematu i celów zajęć.

3. Ćwiczenie nr 1:

powtórzenie *Scratch* - uczniowie wyszukują i definiują w aplikacji takie elementy jak:

- scena,
- obszar budowania skryptów,
- obszar wyboru i edycji „duszków”,
- zakładki Dźwięki, Kostiumy Skrypty,
- lista wyboru kategorii,
- informacje o pozycji X/Y „duszka” na scenie.

4. Wytworzenie sytuacji inspirującej aktywność uczniów – konstruowanie gry w aplikacji *Scratch* do podanego warunku:

Podczas gry dwóch zawodników jednocześnie będzie poruszało swoimi paletkami. Gracze odbijają piłeczkę, pojawiającą się na ekranie w stronę przeciwnika. Wygrywa gracz, który odbije piłeczkę w taki sposób, że ominie ona paletkę przeciwnika, a ten nie zdoła jej odbić.

5. Ćwiczenie nr 2:

dyskusja moderowana przez prowadzącego w zakresie doboru obiektów oraz elementów warunkujących sterowanie zarówno przez program jak i ucznia.

6. Ćwiczenie nr 3:

Burza mózgów:

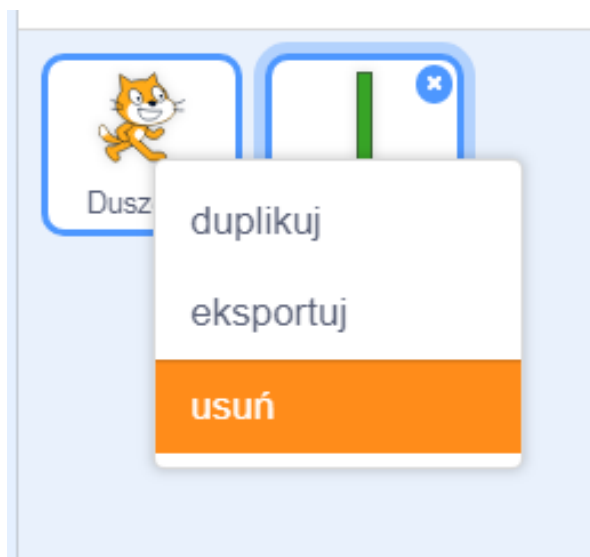
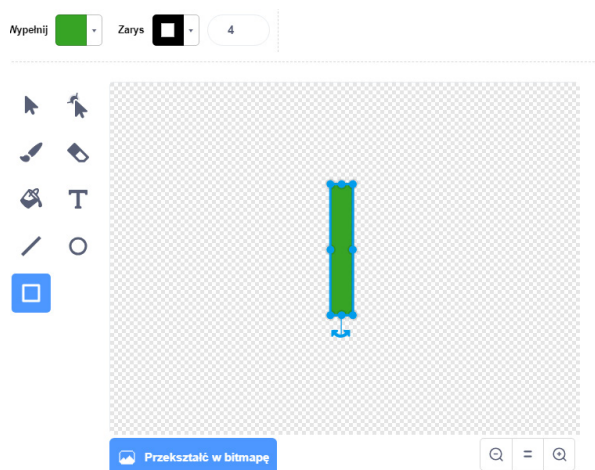
Sposoby sterowania w grach komputerowych - przykładowe odpowiedzi: joystick, myszka, klawiatura. Podsumowanie z omówieniem zastosowań przykładowych rozwiązań. Argumentacja zasadności wykorzystania w projektowanej grze klawiatury komputerowej.

7. Instruktaż prowadzącego do schematu programowania z uwzględnieniem trzech głównych kategorii:

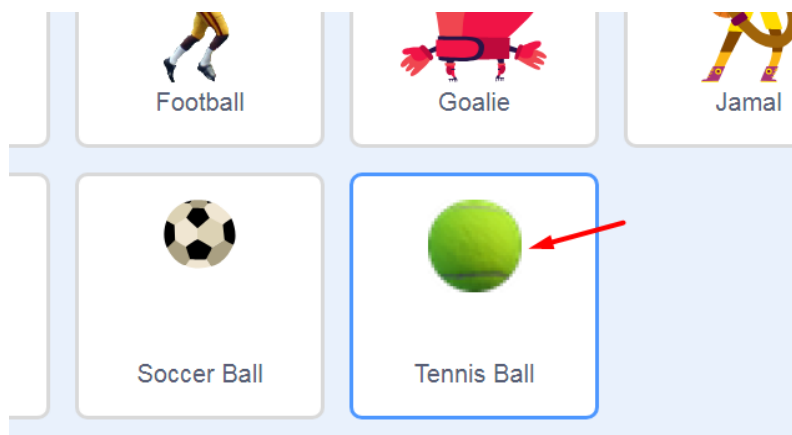
- ustawienia początkowe,
- sterowanie grą,
- akcje.

8. **Ćwiczenie nr 4** - Praca w parach – projektowanie gry *Ping pong* w aplikacji *Scratch* wg. wyznaczonych etapów pracy.

- projektowanie nowego „duszka” (paletka) niezbędnego do zaprogramowania gry oraz usuwanie domyślnego „duszka” - kotka,

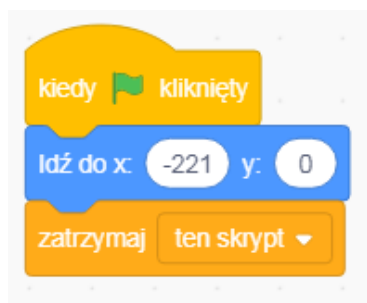


- powielanie „duszka” – paletki z wykorzystaniem funkcji DUPLIKUJ,
- dodawanie nowego „duszka” –piłeczki - korzystając z BIBLIOTEKI programu,
- nadawanie „duszkom” nazwy (paletkom i piłce),
- projektowania sterowania paletkami poprzez wprowadzenie współrzędnych paletek i piłki po uruchomieniu gry.

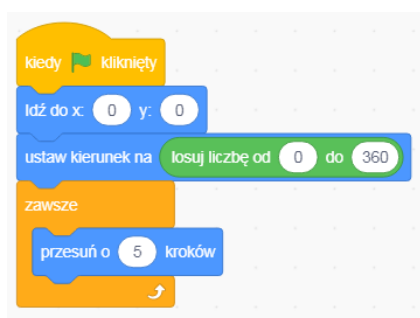


Przykładowy skrypt z wprowadzonymi współrzędnymi startowymi dla lewej paletki:

- tworzenie skryptów dla obiektów – sterowanie paletkami,
- testowanie poprawności sterowania paletkami za pomocą klawiatury,
- tworzenie skryptu dla obiektu – piłeczki.



Przykładowy skrypt:



- tworzenie skryptów dla zaplanowanych akcji dotyczących:
 - odbicia piłeczki od krawędzi,
 - piłeczka dotyka paletki,
 - piłeczka omija paletkę,
- uzupełnianie uprzednio stworzonych skryptów o niezbędne instrukcje warunkowe, wyświetlanie napisów oraz dodawanie czujników,
- tworzenie zmiennych o przykładowych nazwach: licznik prawy, licznik lewy,
- testowanie w parach zaprogramowanej przez siebie gry.

9. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 10

czas trwania: 2 godz.

Budowa dźwigu *LEGO Mindstorms*.

Cel ogólny:

Przekazanie uczniom wiedzy i umiejętności umożliwiających samodzielnie skonstruowanie działającego modelu dźwigu *LEGO Mindstorms*.

Cele szczegółowe. Uczeń:

- wymienia elementy budowy maszyn prostych i określa ich funkcje (dźwignia, kołowrót, drążek),
- zna budowę i zasadę działania różnych typów dźwigów (m.in. dźwig portowy, żuraw dźwigowy, winda),
- konstruuje z klocków *LEGO Mindstorms EV3* model dźwigu,
- wykorzystuje przekładnię zębatą i wał napędowy do zbudowania napędu ruchomego ramienia żurawia,
- rozwiązuje zadanie problemowe według własnego projektu,
- przestrzega ustalonych norm współpracy w grupie.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- przygotowanie zestawów klocków *Lego Mindstorms EV3* oraz stanowisk z laptopami,
- przygotowanie prezentacji multimedialnej zawierającej modele maszyn prostych,
- wspieranie uczniów w rozwiązywaniu pojawiających się trudności, moderowanie pracy grupy.

Rola uczniów (w wieku 10-18 lat):

- zapoznanie się z instrukcją konstrukcji klocków *Lego Mindstorms EV3*,
- postępowanie zgodne z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianych problemów,
- wyszukiwanie, analiza i selekcjonowanie treści,
- aktywny udział w zajęciach,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- podejmowanie dyskusji w zakresie omawianego problemu.

Przygotowanie do zajęć – środki dydaktyczne:

- prezentacja multimedialna w Power Point,
- rzutnik,
- laptop,
- zestaw *LEGO Mindstorms EV3*,
- oprogramowanie *LEGO Mindstorms Education EV3*.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.
2. Przedstawienie tematu i celów zajęć.
3. Instruktaż połączony z prezentacją multimedialną - działania wybranych maszyn prostych (dźwignia, kołowrót, krążek). Budowy i zasady działania różnych typów dźwigów.
4. Omówienie znaczenia pojęcia związanego z przeciwwagą i równowagą.
5. Projektowanie konstrukcji modelu robota – wg podanej instrukcji (film instruktażowy), uwzględnienie w kolejnym etapie przekładni zębatej do zbudowania napędu ruchomego ramienia żurawia.
6. Prezentacja na forum grupy przygotowanych modeli z komentarzem w zakresie konstrukcji i przeznaczenia.
7. Dyskusja moderowana przez prowadzącego:
 - Z jakich części składa się robot?
 - Jak działa?
 - W jaki sposób przenoszony jest napęd?
 - Jakie mechanizmy zastosowano w modelu?
 - Gdzie znajduje się przeciwwaga?
 - Jak działa wyciągarka robota?
 - Gdzie spotkamy tego typu maszyny?
8. Podsumowanie zajęć.
9. Ewaluacja zajęć z wykorzystaniem techniki „kosz i walizeczka”.

scenariusz 11

czas trwania: 2 godz.

Programowanie modelu dźwigu.

Cel ogólny:

Doskonalenie kompetencji uczniów w samodzielnym programowaniu modeli z wykorzystaniem napędu.

Cele szczegółowe. Uczeń:

- wykorzystuje przyciski *kostki EV3* do zaprogramowania układu sterowania,
- stosuje pętlę i instrukcję warunkową w zadaniu programistycznym,
- poznaje wszystkie elementy i możliwości *kostki EV3*,
- rozwiązuje zadanie problemowe wg własnego projektu,
- przestrzega ustalonych norm współpracy w grupie.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie zestawów klocków *Lego Mindstorms EV3* oraz stanowisk z laptopami,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- opracowanie prezentacji multimedialnej - zagadnienia związane z wykorzystaniem *kostki EV3*.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodne z wyznaczonymi etapami pracy,
- włączanie się do dyskusji w zakresie omawianych problemów,
- aktywny udział w zajęciach,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- wyszukiwanie informacji z zakresu programowania poprzez analizę instrukcji.

Przygotowanie do zajęć – środki dydaktyczne:

- prezentacja multimedialna,
- laptop,
- projektor multimedialny,
- zestaw *LEGO Mindstorms* wraz z *kostką EV3*,
- oprogramowanie *LEGO Mindstorms Education EV3*.



Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.
2. Przedstawienie uczniom tematu i celów zajęć.
3. **Ćwiczenie nr 1:** programowanie modelu przy asyście prowadzącego. Poznanie możliwości pracy liny oraz ramienia dźwigu.
4. Omówienie pojęcia instrukcji warunkowej.
5. Instruktaż z prezentacją sposobu zastosowania instrukcji warunkowej w zakresie uniezależnienia działania robota od stanu klawiszy na kostce EV3.
6. Prezentacja programu modelu z uwzględnieniem zwolnienia przycisku.
7. **Ćwiczenie nr 2:** praca w grupach – opracowanie infografiki – budowa, programowanie i testowanie robota. Gdzie można spotkać odpowiedniki zbudowanego modelu?
8. Prezentacja powstałych projektów w wersji elektronicznej.
9. Podsumowanie zajęć. Runda w kręgu – programowanie jest dla mnie...
10. Ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 12

czas trwania: 2 godz.

LEGO Mindstorms – budujemy robota gąsiennicowego.

Cel ogólny:

Nabycie podstawowej wiedzy z zakresu funkcjonalności robota i jego podzespołów. Zdobywanie umiejętności pozwalających skonstruować robota.

Cele szczegółowe. Uczeń:

- ma pogłębioną wiedzę z zakresu funkcjonalności robota i jego podzespołów,
- podaje przykłady automatyzacji czynności życia codziennego przez roboty,
- wyróżnia roboty programowalne i humanoidalne,
- identyfikuje elementy elektroniczne robota stanowiące o jego funkcjonalności,
- uruchamia robota,
- tworzy dla robota tor przeszkód/trasę, którą ma pokonać za pomocą zdalnego sterowania,
- współpracuje w grupie zgodnie z przyjętymi standardami,
- wyszukuje różne drogi rozwiązania sytuacji problemowej.

Rola Prowadzącego:

- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszający, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie zestawów klocków *LEGO Mindstorms EV3* oraz stanowisk z laptopami,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- moderowanie zadawanych pytań naprowadzających uczniów na skuteczne rozwiązanie postawionego problemu,
- przygotowanie prezentacji multimedialnej zawierającej filmy instruktażowe:

> <https://youtu.be/FYaF4nik08Y?t=23>

> <https://www.youtube.com/watch?v=rVlhMGQgDkY>

> <https://www.youtube.com/watch?v=Sher6xMGNLg>

> <https://www.youtube.com/watch?v=wckYmiZxPZs>

> <https://www.youtube.com/watch?v=tRdgzllKI58>

• wydzielenie grup zadaniowych o różnym poziomie zaawansowania uczniów,

- budowanie z instrukcji z zestawu LEGO Mindstorms EV3 robota - łaźnika o napędzie gąsiennicowym,
- przygotowanie miejsca, po którym będzie się przemieszczał robot: toru przeszkód.

Rola uczniów (w wieku 10-18 lat):

- wykazywanie dbałości o zachowanie porządku na stanowisku pracy,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT,
- organizowanie współpracy w zespole poprzez promowanie pozytywnych działań na rzecz osób z niepełnosprawnościami,
- dokonywanie analizy problemu i aktywny udział w procesie jego rozwiązywania,

Przygotowanie do zajęć – materiały, urządzenia, oprogramowanie:

- dokonywanie wyboru najefektywniejszej drogi rozwiązania sytuacji problemowej.

Przygotowanie do zajęć – materiały, urządzenia, oprogramowanie:

zestaw *LEGO Mindstorms EV3*, laptop, projektor, przeszkody dla robota.

Przed rozpoczęciem zajęć należy pamiętać o doładowaniu akumulatorów w kostce sterującej, by nie wyczerpały się podczas zajęć.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno-porządkowe.
2. Przedstawienie tematu i celów zajęć.
3. Prezentacja multimedialna z wykorzystaniem powyżej wykorzystanych materiałów z serwisu *YouTube*.
4. **Ćwiczenie nr 1** - dyskusja moderowana w zakresie automatyzacji czynności życia codziennego z wykorzystaniem robotów.
5. Demonstracja modelu robota wraz z opisem jego funkcjonalności.
6. **Ćwiczenie nr 2** - praca w grupach 4-6 osobowych - uczniowie w oparciu o instrukcję tworzą konstrukcyjne rozwiązania budowy robota.
7. **Ćwiczenie nr 3** - programowanie urządzenia. Lista zadań do wykonania – prowadzący prezentuje uczniom treści poleceń, jakie mają do wykonania w poznanym środowisku programistycznym. Przykładowe zadania:
 - robot ma pojechać do przodu przez 100 jednostek odległości , następnie ma zawrócić i wrócić na swoje miejsce,
 - robot ma „narysować” kwadrat,
 - robot ma „narysować” kwadrat i na każdym z wierzchołków ma się zatrzymać na 3 sekundy,
 - robot ma „narysować” trójkąt – na każdym z wierzchołków trójkąta ma wydać z siebie dźwięk i wyświetlić dowolny obrazek (każdy obrazek ma być dobrze wyświetlony),
8. **Ćwiczenie nr 4 (poziom zaawansowany)** – uczeń dokonuje zmian w kodzie programów według autorskiej instrukcji do ćwiczenia nr 3.
9. **Ćwiczenie nr 5** - praca w grupach dwuosobowych: zabawa z elementami rywalizacji - „*wyścig robota*” – prowadzący zwraca uwagę na przestrzeganie zasad fair play.
10. Podsumowanie i ewaluacja zajęć:
 - zebranie i uporządkowanie istotnych informacji – dokończ zdanie: Wiem, że robot...
 - ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 13

czas trwania: 2 godz.

Na czym polega programowanie tekstowe w *Python*'ie.

Cel ogólny:

Nabywanie wiedzy i umiejętności z zakresu programowania tekstowego w języku *Python*.

Cele szczegółowe. Uczeń:

- wie do czego służy aplikacja *Python*,
- szuka rozwiązań problemów programistycznych w aplikacji *Python*,
- pisze polecenia w trybie interaktywnym i weryfikuje poprawność działania programu,
- programuje polecenia w trybie interaktywnym *Python Shell* i sprawdza działanie programu,
- pisze program sumujący dwie liczby w trybie skryptowym,
- wpisuje odpowiednie polecenia w aplikacji *Python*, aby osiągnąć założony efekt końcowy np. do obliczania pola i obwodu kwadratu.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- moderowanie zadawanych pytań naprowadzających uczniów na skuteczne rozwiązanie postawionego problemu,
- dostosowanie stopnia trudności zadań do uczniów ze specjalnymi potrzebami edukacyjnymi.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodnie z zaproponowanymi etapami pracy,
- podejmowanie dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery/laptopy z oprogramowaniem i dostępem do Internetu,
- karty pracy.

Przebieg zajęć:

1. Czynności organizacyjno- porządkowe.
2. Przedstawienie tematu i celów zajęć.
3. **Ćwiczenie nr 1** - praca w zespołach dwuosobowych - wyszukanie w Internecie definicji programowania tekstowego w języku Python. Prezentowanie przez uczniów wybranych definicji ze wskazaniem źródła. Porównanie brzmienia definicji zawartej w zasobach *Wikipedii* (<https://pl.wikipedia.org/wiki/Python>). Interpretacja treści definicji programowania tekstowego w języku *Python* przez uczniów.
4. **Ćwiczenie nr 2** - *Burza mózgów* – stworzenie przestrzeni do dyskusji dydaktycznej - uczniowie w grupach 3 osobowych przygotowują plakat graficzny pt. „Na czym polega programowanie tekstowe *Python’ie?*” Prezentacja przez liderów grup i podsumowanie przedstawionych propozycji przez prowadzącego.
5. **Ćwiczenie nr 3** - programowanie tekstowe w aplikacji *Python*, uczniowie w parach programują polecenia w trybie interaktywnym i sprawdzają działanie programu oraz opracowują program sumujący dwie liczby w trybie skryptowym.
6. Wyjaśnienie prowadzącego dotyczące programowania tekstowego:
 - *Python* jest jednym z tekstowych języków programowania, wśród których można wymienić m.in. *JavaScript*, *C++*, *HTML*,
 - można go pobrać ze strony python.org, korzystając z zakładki *Downloads*,
 - rozpowszechniany jest jako otwarte oprogramowanie, a jego licencja pozwala na nieodpłatne użytkowanie,
 - programy zapisywane są w edytorze kodu *IDLE Python*,
 - tryb interaktywny w powłoce *Python Shell* pozwala na wpisywanie poleceń, a tryb skryptowy – na zapisywanie ich do pliku.
7. Prowadzący rozdaje uczniom kartki i prosi o uzupełnienie wy kropkowanych fragmentów wyrazami:

Python • tekstowych • poleceń • klawiatury • skryptami • otwarte • licencja

.....jest jednym zjęzyków programowania, w którym wiersze.....wpisywane są za pomocąTworzone programy nazywane sąRozpowszechniany jest jako oprogramowanie , a jego pozwala na nieodpłatne użytkowanie.

8. Prowadzący pokazuje i objaśnia:
 - w jaki sposób napisać polecenia w trybie interaktywnym (np. https://www.onlinegdb.com/online_python_interpreter) i sprawdzić działanie programu.

Prowadzący zwraca uwagę na następujące fakty:

- wpisane w oknie *Python Shell* polecenie jest od razu wykonywane,
- tryb interaktywny wykorzystywany jest m.in. do wykonywania obliczeń matematycznych,
- tryb skryptowy służy do tworzenia programów, ich edycji, zapisywania i otwierania,
- zapisane pliki mają rozszerzenie **.py**.
- jakie stosuje się operatory arytmetyczne:

Operator arytmetyczny	Działanie
y	
*	mnożenie
//	dzielenie liczb całkowitych
/	dzielenie liczb ułamkowych (z ułamkiem dziesiętnym)
%	dzielenie liczb całkowitych z resztą (modulo)
**	potęgowanie

- jak napisać program sumujący dwie liczby w trybie skryptowym:

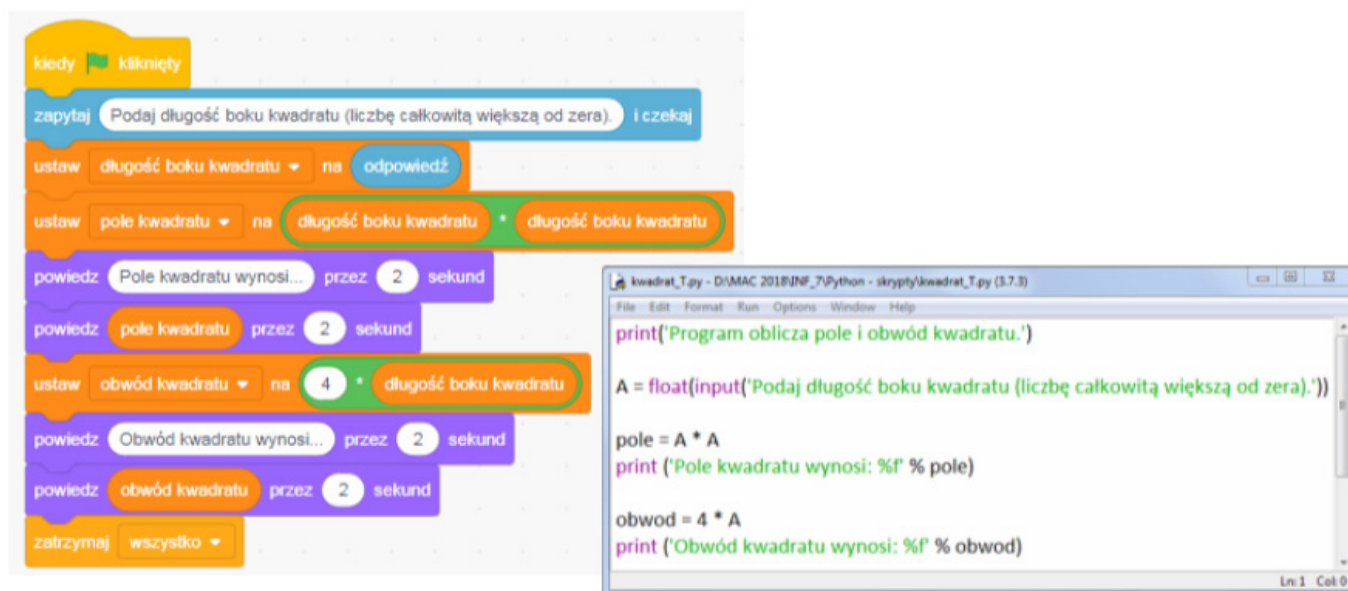
Prowadzący zwraca uwagę na fakt, że:

- instrukcja wyjścia *print* wyświetla dane wejściowe lub wyjściowe jako ciąg znaków (tekst) zapisany w nawiasach po między apostrofami (` `),
- zmienne A i B reprezentują dane wejściowe,
- instrukcja wejścia *input* pobiera ciąg znaków (tekst) wpisany za pomocą klawiatury,
- funkcja *float* zamienia wpisany ciąg znaków na liczbę,
- zmienna suma oblicza sumę danych wejściowych, liczb A i B,
- znak % informuje o miejscu wstawienia wartości zmiennej,
- litera *f* określa typ danych jako liczbę rzeczywistą.

9. Ćwiczenie nr 4 - uczniowie w parach uruchamiają aplikację *Python*, poznają okno programu i programują algorytm sumowania dwóch liczb w trybie interaktywnym *Python Shell* i w trybie skryptowym oraz sprawdzają działanie programu.

10. Prowadzący przygotowuje zadania o różnym stopniu trudności dostosowane dla uczniów ze specjalnymi potrzebami edukacyjnymi. Prowadzący udziela uczniom spersonalizowanego wsparcia.

11. Ćwiczenie nr 5 - uczniowie porównują skrypty utworzone za pomocą wizualnego języka programowania *Scratch* i tekstowego – *Python*. Określają podobieństwa i różnice? Wprowadź podany kod źródłowy w trybie skryptowym, aby



sprawdzić działanie programu.

Przykładowy kod programu w *SCRATCH* i *PYTHON*IE

12. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji z zakresu programowania tekstowego,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 14

czas trwania: 2 godz.

Instrukcje warunkowe, operatory logiczne i operatory porównania w języku Python.

Cel ogólny:

Nabycie przez uczniów wiedzy i umiejętności w zakresie instrukcji warunkowych oraz właściwych operatorów logicznych i operatorów porównania.

Cele szczegółowe. Uczeń:

- zna instrukcje warunkowe (*if*, *if-else*, *if-elif-else*) w języku Python,
- potrafi uzupełnić program instrukcją dzielenia liczb,
- zna operatory logiczne (*and*, *or*) i operatory porównania (*==*, *!=*, *<*, *>*, *<=*, *>=*),
- przestrzega zasady precyzyjnego formułowania komunikatów,
- podejmuje współpracę w grupie zgodnie z ustalonymi zasadami.

Rola prowadzącego:

- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedośłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- dostosowanie stopnia trudności zadań dla uczniów ze specjalnymi potrzebami edukacyjnymi.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodne z zaproponowanymi etapami poszczególnych zadań,
- podejmowanie dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery z oprogramowaniem i dostępem do Internetu,
- projektor multimedialny.

Opis zadań i sposób ich realizacji:

1. Czynności organizacyjno - porządkowe.
2. Przedstawienie tematu i celów zajęć.
3. Wytworzenie sytuacji sprzyjającej aktywności edukacyjnej uczniów – wspólna analiza wypowiedzi L. Torvaldsa (twórcy systemu operacyjnego Linux): „Większość programistów programuje nie dlatego, że spodziewają się zapłaty lub uwielbienia tłumów, ale dlatego, że programowanie jest dla nich zabawą”.
4. **Ćwiczenie nr 1:** uczniowie korzystając z Internetu przygotowują informacje na temat programowania tekstowego w języku *Python* (<https://rk.edu.pl/pl/wprowadzenie-do-pythona/>)
5. **Ćwiczenie nr 2:** na czym polega programowanie tekstowe w języku *Python* z wykorzystaniem instrukcji warunkowych, właściwych operatorów logicznych i operatorów porównania?
6. Podsumowanie przygotowanych przez uczniów informacji.
7. Prowadzący uzupełnia informację dotyczące programu, objaśnia instrukcję dzielenia liczb. Zadaniem uczniów jest napisanie według wzoru, pod iloczynem kolejnych wierszy kodu (instrukcję warunkową) - (tak aby program wykonywał dzielenie).
8. Omówienie kolejnych kroków tworzenia programu do wykonywania obliczeń. Zadaniem uczniów jest:
 - otworzyć program, według wzoru napisać kolejne wiersze kodu (instrukcję warunkową), tak aby program wykonywał dzielenie,
 - zastosować instrukcję (*if*) i instrukcję (*else*) oraz operatory porównania,
 - zapisać program i sprawdzić jego działanie.

```

dzialania_arytmetyczne.py
File Edit Format Run Options Window Help
print ("Program służy do wykonywania obliczeń.")
A = float(input("Podaj pierwszą liczbę. A = "))
B = float(input("Podaj drugą liczbę. B = "))
suma = A + B
print ('Suma wynosi %f' %(suma))
roznica = A - B
print ('Różnica wynosi %f' %(roznica))
iloczyn = A * B
print ("Iloczyn wynosi %f" %(iloczyn))
if B!=0:
    iloraz = A / B
    print ("Iloraz wynosi", iloraz)
else:
    print ("Błąd dzielenia. Nie dziel przez zero!")
Ln:15 Col:0
    
```

Operator logiczny	Działanie
and (i) koniunkcja	spełnione muszą być oba podane warunki
or (lub) alternatywa	musi być spełniony jeden z dwóch warunków
Operator porównania	Działanie
==	jest równe
!=	nie jest równe
<	mniejsze
>	większe
<=	mniejsze lub równe
>=	większe lub równe

9. **Ćwiczenie nr 3:** Prowadzący inicjuje pracę w grupach. Uczniowie tworzą grupy 2-3 osobowe i wykonują ćwiczenia:

- $0 > 1, 0 < 1, 0 \leq 1, 1 == 0, 0 \geq 1, 1 == 1, 1 != 0, 1 != 1$ (grupa I),
- $x = 100, x < 90, x > 10, x > 1$ and $x < 101, x < 1$ and $x < 101, x > 1$ or $x < 101, x != 100$ or $x > 10, x > 10$ and $x < 110$ or $x != 100$ (grupa II).

10. **Ćwiczenie nr 4:** Prowadzący rozdaje uczniom kartki i prosi o uzupełnienie wykropkowanych fragmentów wyrazami:

if • warunku • if-else • spełniony • wykonana • przeciwnym razie
instrukcja • If-elif-else • złożona • warunkowa • warunków

Instrukcja..... (jeżeli...) może być wykonana po spełnieniu podanego..... Instrukcja(jeżeli..., w przeciwnym razie...) ma zastosowanie w sytuacji, gdy warunek jesti instrukcja zostanie....., w – zostanie wykonana kolejna..... Instrukcja(jeżeli..., jeżeli natomiast..., w przeciwnym razie...) toinstrukcja....., dzięki której można sprawdzać wiele

11. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji z zakresu praktycznego stosowania instrukcji warunkowych, operatorów logicznych i operatorów porównania,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

scenariusz 15

czas trwania: 2 godz.

Programowanie i testowanie algorytmów w języku *Python*.

Cel ogólny:

Nabycie przez uczniów wiedzy i umiejętności programowania tekstowego w języku *Python* z wykorzystaniem instrukcji warunkowych oraz właściwych operatorów logicznych i porównania.

Cele szczegółowe. Uczeń:

- potrafi napisać program, który:
 - sprawdzi, czy podana liczba jest dodatnia, ujemna czy równa zero,
 - obliczy pole i obwód prostokąta,
- wie, jak porównać skrypty utworzone w języku *Python*,
- dostrzega podobieństwa i różnice w różnych kodach źródłowych programów,
- prezentuje na forum grupy własny projekt.

Rola prowadzącego:

- przedstawienie założonych celów w sposób zrozumiały dla uczniów,
- przygotowanie przestrzeni edukacyjnej sprzyjającej efektywnej pracy uczniów w tym uczniów z niepełnosprawnościami (m.in. niedowidzący, niedosłyszący, niepełnosprawność ruchowa, niepełnosprawność intelektualna w stopniu lekkim),
- przygotowanie stanowisk komputerowych z dostępem do Internetu,
- przygotowanie kart pracy,
- udzielanie wsparcia uczniom wymagającym pomocy psychologiczno- pedagogicznej,
- inspirowanie uczniów do wyrażania swojej opinii związanej z pracą z nowoczesnymi technologiami na forum grupy z przełamaniem barier psychicznych,
- wspieranie uczniów w rozwiązywaniu pojawiających się trudności, moderowanie pracy grupy.

Rola uczniów (w wieku 10-18 lat):

- postępowanie zgodne z zaproponowanymi etapami poszczególnych zadań,
- podejmowanie dyskusji w zakresie omawianego problemu,
- aktywny udział w zajęciach,
- przestrzeganie zasad bezpiecznego korzystania z technologii IT.

Przygotowanie do zajęć – środki dydaktyczne:

- komputery z oprogramowaniem i dostępem do Internetu,
- karty pracy „prawda- fałsz”.

Opis zadań i sposób ich realizacji:

1. **Ćwiczenie nr 1:** uczniowie w parach programują w *Pythonie* według podanych instrukcji:

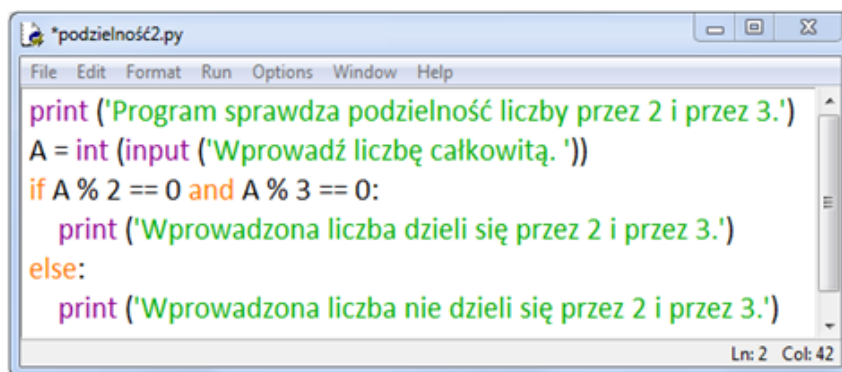
- Napisz program, który sprawdzi, czy podana liczba jest dodatnia, ujemna czy jest równa zero. W oknie *Python Shell* uruchom nowy plik. W oknie edytora *Untitled* wprowadź kod źródłowy programu według wzoru. Zapisz program pod nazwą „jeżeli” i sprawdź jego działanie.
- Napisz program, który obliczy pole i obwód prostokąta. Zapisz program pod nazwą „prostokąt”. Zwróć uwagę, że wprowadzone dane muszą być liczbami całkowitymi. Następnie wprowadź instrukcję (*if/else*), która uwzględni warunek, aby wprowadzone dane (długość obu boków prostokąta) były większe od zera.
- Sprawdź działanie programu.

2. Prowadzący monitoruje pracę uczniów i udziela stosownego wsparcia.

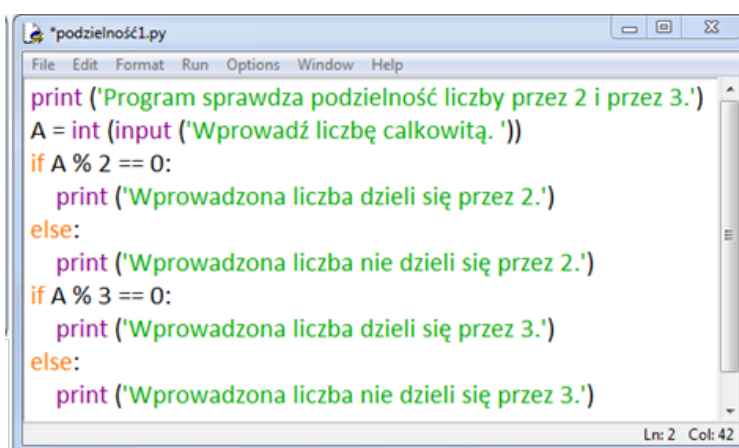
3. Komentarz prowadzącego - wiersz z instrukcją warunkową (*if*) zawiera wyrażenie logiczne (*and*), które oznacza, że jeżeli obydwie zmienne A i B spełnią warunek (będą większe od zera), to program wyświetli wyniki obliczeń. W przeciwnym razie (*else*) wyświetlony zostanie komunikat o konieczności wprowadzania liczb dodatnich.

4. **Ćwiczenie nr 2:** uczniowie porównują skrypty utworzone w języku *Python*, które sprawdzają, czy wprowadzona liczba jest podzielna przez liczbę 2 i liczbę 3. Wskazują różnice w kodach źródłowych obydwu programów.

5. **Ćwiczenie nr 3:** uczniowie pracują w parach, wprowadzają po jednym kodzie w trybie skryptowym, aby sprawdzić działanie programów.



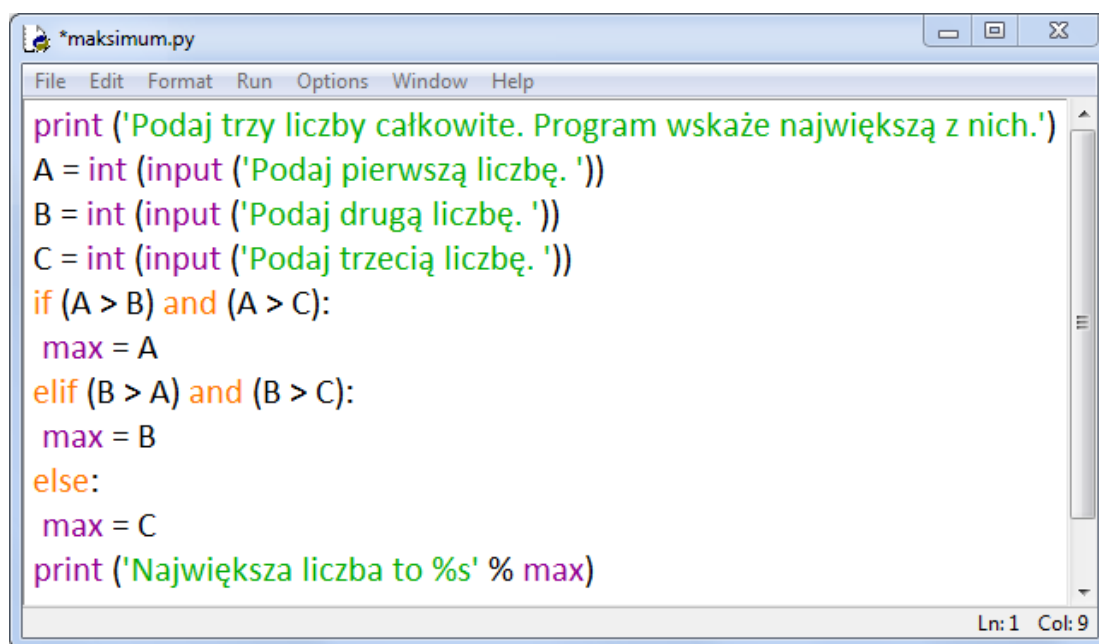
```
*podzielność2.py
File Edit Format Run Options Window Help
print ("Program sprawdza podzielność liczby przez 2 i przez 3.")
A = int (input ("Wprowadź liczbę całkowitą. "))
if A % 2 == 0 and A % 3 == 0:
    print ("Wprowadzona liczba dzieli się przez 2 i przez 3.")
else:
    print ("Wprowadzona liczba nie dzieli się przez 2 i przez 3.")
Ln: 2 Col: 42
```



```
*podzielność1.py
File Edit Format Run Options Window Help
print ("Program sprawdza podzielność liczby przez 2 i przez 3.")
A = int (input ("Wprowadź liczbę całkowitą. "))
if A % 2 == 0:
    print ("Wprowadzona liczba dzieli się przez 2.")
else:
    print ("Wprowadzona liczba nie dzieli się przez 2.")
if A % 3 == 0:
    print ("Wprowadzona liczba dzieli się przez 3.")
else:
    print ("Wprowadzona liczba nie dzieli się przez 3.")
Ln: 2 Col: 42
```

6. Ćwiczenie nr 4 (dla uczniów zaawansowanych):

Przeanalizuj kod źródłowy programu *maksimum*, który wskazuje największą spośród trzech podanych liczb. Opisz go słowami. Napisz program, który wyświetla największą liczbę spośród pięciu podanych. Sprawdź jego działanie.



```
File Edit Format Run Options Window Help
print ('Podaj trzy liczby całkowite. Program wskaże największą z nich.')
A = int (input ('Podaj pierwszą liczbę. '))
B = int (input ('Podaj drugą liczbę. '))
C = int (input ('Podaj trzecią liczbę. '))
if (A > B) and (A > C):
    max = A
elif (B > A) and (B > C):
    max = B
else:
    max = C
print ('Największa liczba to %s' % max)
Ln: 1 Col: 9
```

7. Ćwiczenie nr 5: Uzupełnienie karty pracy „prawda-falsz”.

Czy poniższe zdania są prawdziwe, czy fałszywe?	prawda	fałsz
W języku programowania Python instrukcja warunkowa (if-else) wykonywana jest w przypadku spełnienia określonego warunku.	X	
Podczas tworzenia skryptów za pomocą tekstowego języka programowania konieczna jest znajomość jego składni (reguł).	X	
Operatory służą tylko do zapisywania działań arytmetycznych.		X
Operatory logiczne (and, or) wykorzystywane są w instrukcjach złożonych.	X	
W instrukcjach złożonych należy sprawdzić jeden warunek.		X
Operatory and or nazywane są logicznymi.	X	
Podczas wpisywania instrukcji w kodzie programu należy pamiętać o wcięciach, które są obowiązkowe.	X	

8. Podsumowanie i ewaluacja zajęć:

- zebranie i uporządkowanie istotnych informacji z zakresu praktycznego stosowania instrukcji warunkowych, operatorów logicznych i operatorów porównania,
- ewaluacja zajęć z wykorzystaniem nowych technologii.

NOTATKI

NOTATKI

NOTATKI

