Mikrobitowcy

Autorzy: Dawid Osuchowski, Filip Kłębczyk

Lekcja 5:
Micro:bit i pierwszy program w Pythonie

Podczas zajęć uczniowie zapoznają się z płytką BBC Micro:bit i jej elementami. Podłączą ją do komputera, by zobaczyć ją w działaniu. Napiszą także prosty testowy program, który prześlą na Micro:bita. Na końcu zaś, korzystając z wyświetlacza diodowego, nauczą się wyświetlać i tworzyć własne obrazki.

Cele lekcji:
Uczeń powinien:
- Znać pojęcia: wyświetlacz, środowisko programistyczne, fleszowanie
- Wiedzieć, jak podłączyć płytkę Micro:bit do komputera
- Wiedzieć, jak umieścić na urządzeniu program napisany w edytorze
- Wiedzieć, jak wyświetlić tekst i obrazek

Materiały pomocnicze:
- **http://microbit.org/**
- **https://bbcmicrobitmicropython.readthedocs.io/**
- **https://github.com/MicrobitPolska/FunWithMicrobit**

Pojęcia kluczowe:
→ Python → Micro:bit → fleszowanie → wyświetlacz → środowisko programistyczne

Czas realizacji: 45 min.

Metody pracy:
- wykład,
- dyskusja,
- ćwiczenia praktyczne przy komputerze,
- prezentowanie efektów pracy,
- szukanie rozwiązań na postawiony problem.

Treści programowe:
Podstawa programowa kształcenia ogólnego dla szkół podstawowych – II etap edukacyjny – klasy VII-VIII, informatyka:
I. Rozumienie, analizowanie i rozwiązywanie problemów. Uczeń:
3) przedstawia sposoby reprezentowania w komputerze wartości logicznych, liczb naturalnych (system binarny), znaków (kody ASCII) i tekstów; formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) i wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci schematów blokowych, listy kroków;
5) prezentuje przykłady zastosowań informatyki w innych dziedzinach, w zakresie pojęć, obiektów oraz algorytmów.
II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych. Uczeń:
1) projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia/wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice.
2) projektuje, tworzy i testuje oprogramowanie sterujące robotem lub innym obiektem na ekranie lub w rzeczywistości.
III. Posługiwanie się komputerem, urządzeniami cyfrowymi i sieciami komputerowymi. Uczeń:
2) rozwija umiejętności korzystania z różnych urządzeń do tworzenia elektronicznych wersji tekstów, obrazów, dźwięków, filmów i animacji;
3) poprawnie posługuje się terminologią związaną z informatyką i technologią.
IV. Rozwijanie kompetencji społecznych. Uczeń:
1) bierze udział w różnych formach współpracy, jak: programowanie w parach lub w zespole, realizacja projektów, uczestnictwo w zorganizowanej grupie uczących się, projektuje, tworzy i prezentuje efekty wspólnej pracy;

Przygotowanie przed zajęciami

Jeśli nauczyciel nie zrealizował pierwszych czterech lekcji, powinien przygotować:

1. Oprogramowanie - instalacja na wszystkich komputerach edytora Mu , sprawdzenie czy edytor się uruchamia i czy jest możliwe sfleszowanie (zapisanie pamięci flash) płytki Micro:bit. Dla wygody, by nie usuwać programu demonstracyjnego na wszystkich płytkach, do celów testowych najlepiej wykorzystywać tylko jedną płytkę.

2. Sprzęt - sprawdzenie czy po podłączeniu płytek Micro:bit uruchamia się program demonstracyjny. Jeśli został już nadpisany innym programem (podczas testów lub wcześniejszych zajęć), należy go przywrócić zgodnie z instrukcją zawartą na końcu tego dokumentu.

Wprowadzenie w tematykę i integracja grupy (10 min.)

Jeśli nauczyciel zrealizował Lekcje 1-4 ("Pierwsze kroki z Pythonem"), informuje uczniów, że dziś zaczną programować pytkę Micro: bit. Jeśli nie zrealizował lekcji 1-4, konieczne będzie krótkie wprowadzenie:

Nauczyciel pyta uczniów, czy i jakie języki programowania znają, a o jakich tylko słyszeli. Następnie pyta o typy aplikacji i różne typy urządzeń, na których można je uruchamiać (laptop, telefon komórkowy, telewizor itp.). Zachęcamy do dyskusji na temat tego, czym różnią się aplikacje uruchamiane na różnych urządzeniach?

Opowiada krótko o języku Python, jego zastosowaniach, podaje przykłady popularnych firm oraz aplikacji wykorzystujących język Python (Google, Disney, YouTube, Facebook, Instagram, Dropbox, Spotify).

Język Python jest językiem programowania ogólnego przeznaczenia, z dobrze rozbudowaną biblioteką standardową (określaną często jako “batteries included”, co oznacza, że bez instalacji dodatkowych, zewnętrznych modułów jesteśmy w stanie bardzo dużo zrobić). Dodatkowo jego bardzo prosta i czytelna składnia, pozwala na szybkie rozpoczęcie pracy z językiem, stąd Python jest coraz popularniejszy w zastosowaniach edukacyjnych. Ma jednak tę przewagę, że przede wszystkim sprawdza się w wielu poważnych zastosowaniach i nie jest językiem stricte edukacyjnym.

Znajomość Pythona jest obecnie pożądana na rynku pracy, a w pewnych dziedzinach (uczenie maszynowe, eksploracja wiedzy z dużej ilości danych) jest on w obecnej chwili językiem wiodącym. W rankingu TIOBE najpopularniejszych języków programowania na świecie znajduje się on na trzecim miejscu, po języku Java i C (dane z września 2018).

Język Python można z powodzeniem wykorzystywać między innymi w tworzeniu aplikacji internetowych, tworzeniu gier komputerowych, zastosowaniach naukowych, administracji systemami, symulacjach, szeroko pojętej inżynierii, produkcji, zarządzaniu.

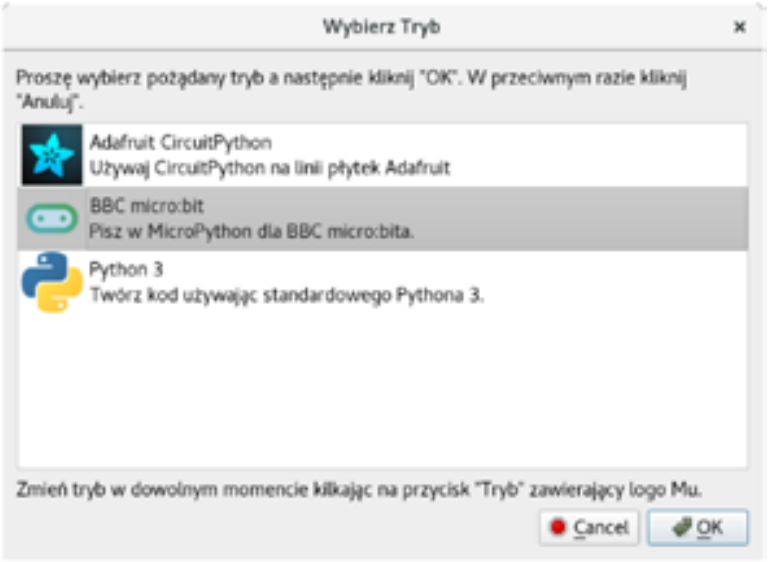
Nauczyciel wskazujemy też na fakt, że Python działa na różnych urządzeniach, w tym na płytkach elektronicznych takich jak Micro:bit, które pokazuje uczniom. Warto też przy tej okazji nadmienić, że jest on jednym z języków do wyboru na maturze z informatyki, więc jego znajomość może się przydać w dalszych etapach edukacji (szkoła średnia).

Na koniec wprowadzenia nauczyciel informujemy uczniów, że przez kolejne zajęcia w ramach programu #SuperKoderzy będą poznawać podstawy języka Python poprzez interakcję z płytką edukacyjną BBC Micro:bit.

Część zasadnicza (30 min.)

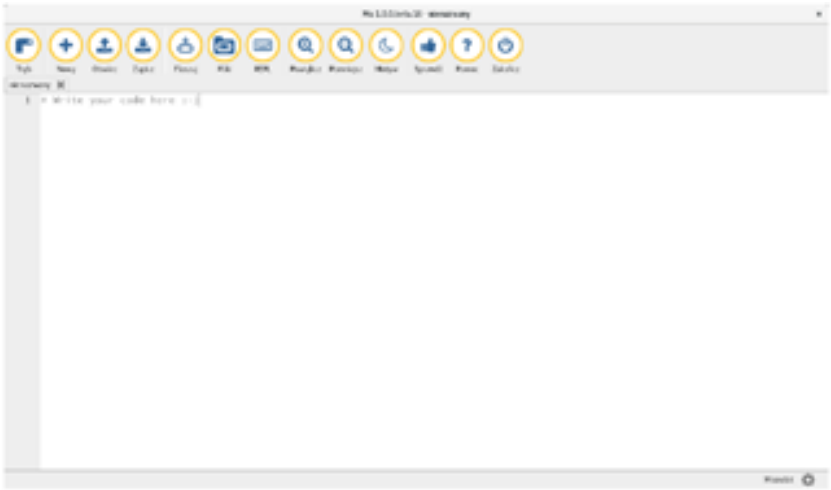
Prosimy uczniów, aby podłączyli płytki Micro:bit do komputera poprzez załączony kabel USB (microUSB), tak jak zaprezentowano na poniższych zdjęciach.

Po podłączeniu do komputera, na urządzeniu powinien rozpocząć działanie program demonstracyjny, za pomocą którego uczniowie są w stanie zapoznać się z podstawowymi elementami płytki, takimi jak wyświetlacz, przyciski oraz akcelerometr, poprzez interaktywne zadania, które muszą wykonać na urządzeniu.



Po tej części, uczniowie uruchamiają edytor Mu. Pierwszym ekranem jest wybór trybu i na potrzeby tych zajęć uczniowie wybierają “BBC micro:bit”. Można przy tej okazji nadmienić (lub przypomnieć), że środowisko programistyczne Mu umożliwia również tworzenie programów na zwykłe komputery z użyciem trybu “Python 3”.

Nauczyciel omawia podstawowe elementy edytora, w szczególności najważniejsze przyciski na pasku (Nowy, Otwórz, Zapisz, Fleszuj, Powiększ, Pomniejsz, Motyw i Sprawdź), ale - na tym etapie - bez demonstrowania ich działania.



Następnie, po przejrzeniu interfejsu przez uczniów, przechodzimy do napisania pierwszego programu. Program jednokrotnie przewinie napis “Ahoj, przygodo!” na wyświetlaczu diodowym.

```
1  from microbit import *
2  display.scroll("Ahoj, przygodo!")
```

Uczniowie za pomocą przycisku «Fleszuj» przesyłają program na płytkę. Podczas tego procesu na tylnej części płytki powinna migać żółta dioda i nie powinniśmy w tym czasie odłączać płytki od komputera.

Po zakończeniu procesu fleszowania program automatycznie uruchamia się na urządzeniu. Aby wykonał się on ponownie, naciskamy przycisk «Reset» (oznaczony na kolejnym zdjęciu), znajdujący się na tylnej części płytki.

Po tym, gdy uczniowie uruchomią już program, nauczyciel objaśnia znaczenie poszczególnych kawałków kodu (zaimportowanie funkcji z modulu microbit i wywołanie funkcji **scroll** z modułu **display**).

Jeśli pozostaje czas, to uczniowie modyfikują program na następujący:

```
1  from microbit import *
2  display.show("Ahoj, przygodo!")
```

Następnie uczniowie odpowiadają, jakie zauważają różnice pomiędzy działaniem funkcji **display.scroll()** i **display.show()**.

W dalszej części nauczyciel omawia budowę diodowego wyświetlacza matrycowego micro:bita i zasady wyświetlania na nim elementów. W kolejnym programie pokazuje uczniom, jak można wyświetlić pojedynczy obrazek. Uczniowie wprowadzają kod, po czym po wpisaniu **Image**. mogą wybrać z listy własny obrazek (np. obrazek serca **Image.HEART**).

```
1  from microbit import *
2  display.show(Image.HEART)
```

Lista wbudowanych obrazków, których można użyć znajduje się w oficjalnej dokumentacji MicroPythona dla BBC Micro:bit: **https://microbit-micropython.readthedocs.io/en/latest/image.html#attributes**

Jeżeli pozostał czas lub uczniowie są ciekawi możliwości tworzenia własnych obrazków, mogą spróbować je zdefiniować w następujący sposób:

```
1  from microbit import *
2
3  box = Image("99999:"
4              "90009:"
5              "90009:"
6              "90009:"
7              "99999")
8
9  display.show(box)
```

Nauczyciel objaśnia, że liczby odpowiadają skali jasności diod wyświetlacza (9 - najjaśniej świecąca, i stopniowo aż do 1 - najciemniej świecąca, 0 - wyłączona).

Wykonanie powyższego kodu na urządzeniu zakończy się rezultatem jak na zamieszczonym zdjęciu.

Poproś uczniów o wykorzystanie kilku różnych wartości jasności przy definiowaniu własnych obrazków.

Podsumowanie (5 min.)

Pytamy uczniów, co najbardziej podobało się im na lekcji i co sprawiło trudność. Do jakich zastosowań wyświetlacz Micro:bita jest niewystarczający?

Uwagi/Alternatywy

Często zdarza się, że Micro:bit na którym pracujemy ma już załadowany jakiś program - czasami zależy nam by po podłączeniu urządzenie nic nie robiło. W tym celu najprościej zapisać pamięć Flash programem pustym (brak linii kodu).

Jeśli chcemy przywrócić program demonstracyjny, należy najpierw pobrać go w formacie .hex pod poniższym linkiem: **https://support.microbit.org/helpdesk/attachments/19033089764**

Następnie podłączamy Micro:bita, który powinien pojawić się w naszym systemie jako pendrive. Kopiujemy plik .hex na urządzenie co wiąże się z załadowaniem programu do pamięci flash i uruchomieniem Micro:bita z tym programem.

Wszystkie programy dla Micro:bita są tworzone z wykorzystaniem MicroPythona. Jest to specjalna wersja Pythona przystosowana do urządzeń o ograniczonych zasobach sprzętowych takich jak pamięć operacyjna czy procesor. W przypadku tworzenia programów na komputerze stacjonarnym lub laptopie wykorzystujemy standardową wersję Pythona (najczęściej jest to implementacja CPython), która oferuje znacznie więcej przydatnych modułów dla programisty. Sama składnia języka jest jednak taka sama, więc nie dysponując Micro:bitem możemy uczyć się tych samych konstrukcji języka programowania. Warto wspomnieć o tej możliwości uczniom, jednocześnie zaznaczając, że będą w inny sposób realizować interakcję z użytkownikiem programu - zamiast modułu **display** obsługującego wyświetlacz diodowy Micro:bita, będą korzystać po prostu z funkcji **print()**, zaś do pobierania danych od użytkownika z funkcji **input()**.

Dla Micro:bita można też tworzyć programy z wykorzystaniem bloczków podobnych do Scratcha z wykorzystaniem narzędzia JavaScript Blocks Editor (**https://makecode.microbit.org/**). Jest to jednak rozwiązanie nie mające związku z Pythonem i nie zalecamy jego wykorzystania w klasach 7-8, ponieważ opóźni to naukę samego Pythona. Zastosowanie tego rozwiązania może być za to dobrym wyborem w klasach młodszych.