

Stanisław SZABŁOWSKI 

ORCID: 0000-0001-7287-8590. Prof. PANS. Dr inż., Państwowa Akademia Nauk Stosowanych w Przemysłu, Wydział Nauk Technicznych i Sztuk Projektowych, Instytut Nauk Technicznych, ul. Żołnierzy I Armii Wojska Polskiego 1E, 37-700 Przemysł, e-mail: s.szablowski@pansp.pl

data złożenia tekstu do Redakcji DI: 19.03.2025; data wstępnej oceny artykułu: 27.03.2025

PROGRAMOWANIE APLIKACJI MOBILNYCH W EDUKACYJNYCH PROJEKTACH MECHATRONICZNYCH

MOBILE APPLICATION PROGRAMMING IN EDUCATIONAL MECHATRONIC PROJECTS

Słowa kluczowe: aplikacja mobilna, programowanie, MIT App Inventor.

Keywords: mobile app, programming, MIT App Inventor.

Streszczenie

W opracowaniu zaprezentowano praktyczne aspekty projektowania aplikacji mobilnych w edukacyjnych systemach mechatronicznych. Przeanalizowano wybrane problemy metodyki i narzędzia programistyczne wspomagające programowanie aplikacji mobilnych. Scharakteryzowano środowisko graficzne MIT App Inventor oraz przeprowadzono analizę i ocenę możliwości uczenia się programowania aplikacji mobilnych. Opisano metodykę Design Thinking projektowania aplikacji na przykładzie edukacyjnego robota mobilnego. W podsumowaniu wskazano środowisko MIT App Inventor jako wiodące narzędzie do uczenia się programowania.

Abstract

The paper presents practical aspects of designing mobile applications in educational mechatronic systems. Selected problems of the methodology and programming tools supporting the programming of mobile applications are analyzed. The MIT App Inventor graphical environment is characterized and the possibilities of learning to program mobile applications are analyzed and assessed. The Design Thinking methodology for application design is described using the example of an educational mobile robot. The summary indicates the MIT App Inventor environment as the leading tool for learning programming.

Wstęp

Aplikacje mobilne są oprogramowaniem zaprojektowanym do działania na urządzeniach przenośnych, takich jak smartfony i tablety. Stały się one integralną częścią naszego życia, rewolucjonizując sposób, w jaki komunikujemy się, pracujemy, uczymy się i spędzamy wolny czas. Aplikacje w telefonach (smartfonach, tabletach) oferują bardzo szeroki wachlarz funkcji i celów, które można dostosować do indywidualnych potrzeb i zainteresowań użytkowników. Obejmują one niemal wszystkie sfery życia – od zabawy i rozrywki po pracę, technikę i edukację. Dzięki ich różnorodności oraz elastycznym funkcjom możemy dostosować urządzenia mobilne do naszych indywidualnych potrzeb. Obecnie obserwujemy nowe trendy w rozwoju aplikacji mobilnych. Należą do nich: wykorzystanie sztucznej inteligencji (AI) do personalizacji doświadczeń użytkownika, analizy danych i automatyzacji procesów oraz technologie rzeczywistości rozszerzonej (AR) i wirtualnej (VR) mające zastosowanie w grach, aplikacjach edukacyjnych i e-commerce.

Programowanie aplikacji mobilnych odgrywa kluczową rolę na wszystkich poziomach edukacji informatyczno-technicznej, wspierając zarówno nauczycieli, jak i uczniów w projektowaniu systemów technicznych. W edukacji technicznej wspiera projekty DIY (ang. Do It Yourself), pozwalając uczniom i studentom na przeprowadzanie własnych eksperymentów i tworzenie oryginalnych projektów. Analiza literatury przedmiotu wskazuje wiele przykładów, w których programowanie, nie tylko aplikacji mobilnych, przyczynia się do efektywnego kształcenia i rozwoju intelektualnego uczących się.

O metodyce projektowania i narzędziach programistycznych

Do tworzenia aplikacji mobilnych wykorzystuje się różne technologie i narzędzia w zależności od ich rodzaju i platformy docelowej. Można stosować różne języki programowania i narzędzia dla wielu systemów operacyjnych. Proces budowy aplikacji obejmuje również ważne etapy, takie jak projektowanie interfejsu użytkownika, testowanie i optymalizację pod kątem wydajności oraz zgodności z różnymi urządzeniami mobilnymi oraz systemami operacyjnymi iOS i Android.

Projektowanie aplikacji mobilnych polega na tworzeniu atrakcyjnych wizualnie, przeznaczonych dla użytkowników programów, które w pełni zaspokajają ich wymagania i oferują przyjemne wrażenia podczas pracy na smartfonie lub tablecie. Wskazane jest stosowanie metodyki Design Thinking¹. Design Thin-

¹ <https://designthinking.pl/co-to-jest-design-thinking/>

king dzięki swojemu uniwersalnemu charakterowi ma szerokie zastosowanie wszędzie tam, gdzie mamy do czynienia z problemami dywergencyjnymi, które posiadają wiele różnych rozwiązań. Projektowanie zazwyczaj obejmuje korzystanie z różnych narzędzi cyfrowych na poszczególnych jego etapach. Narzędzia te mają funkcje, które ułatwiają tworzenie projektów aplikacji, współpracę i prototypowanie, znacznie pomagając projektantom w pracy².

Jednym z popularnych narzędzi jest Android Studio, czyli oficjalne IDE firmy Google do tworzenia aplikacji na Androida, które używa Javy jako podstawowego języka programowania. Android Studio wymaga instalacji w środowiskach kompatybilnych z systemami operacyjnymi Windows, macOS i Linux. Zawiera emulator do testowania aplikacji na różnych urządzeniach i wersjach Androida³.

Podczas projektowania dobrej aplikacji mobilnej należy przestrzegać ustrukturyzowanego procesu, który zapewnia zorganizowany, spójny i skoncentrowany na użytkowniku produkt końcowy. Android stosuje zestaw zasad projektowania o nazwie Material Design, który jest kompleksowym systemem projektowania mającym zastosowanie do wszystkich urządzeń z Androidem. Zapewnia on wytyczne dotyczące projektowania interakcji, wizualizacji, układu i nie tylko. Interfejs użytkownika jest kluczowym elementem aplikacji mobilnej. Musi być prosty, intuicyjny i estetyczny. Należy pamiętać o zachowaniu spójności i jednolitości w projekcie graficznym.

Projektowanie interfejsów użytkownika aplikacji mobilnych staje się łatwiejsze dzięki platformom „No-Code”. Komponenty interfejsu użytkownika na ekranie można ułożyć zgodnie z projektem przez technikę „drag & drop” i dostosować ich wygląd za pomocą różnych opcji stylizacji. Takie podejście eliminuje potrzebę tworzenia projektów w zewnętrznych narzędziach, takich jak Sketch lub Figma i ręcznego wdrażania tych projektów w kodzie⁴.

Bezpłatne narzędzia do tworzenia aplikacji w graficznych środowiskach IDE zyskują na popularności, umożliwiając użytkownikom tworzenie, testowanie i wdrażanie aplikacji bez zaawansowanej wiedzy programistycznej. Platformy te udostępniają interfejsy, szablony i predefiniowane komponenty „drag&drop”, ułatwiając użytkownikom bez przygotowania informatycznego tworzenie i modyfikowanie aplikacji mobilnych⁵. Narzędzia do tworzenia aplikacji mobilnych

² I. Stachowicz, *Projektowanie aplikacji mobilnych – kluczowe zasady*, <https://innovatika.com/pl/projektowanie-aplikacji-mobilnych-kluczowe-zasady>, 2023; G. Szymański, *Projektowanie i reklama aplikacji mobilnych*, Wyd. Stowarzyszenie Innowatorów Zarządzania, Łódź 2021.

³ C. Delessio i in., *Android Studio w 24 godziny. Wygodne programowanie dla platformy Android*, wyd. IV, Helion 2016; M. Płonkowski, *Android Studio. Tworzenie aplikacji mobilnych*, Helion 2017.

⁴ <https://yarmobile.com/pl/projektowanie-aplikacji-mobilnych/>

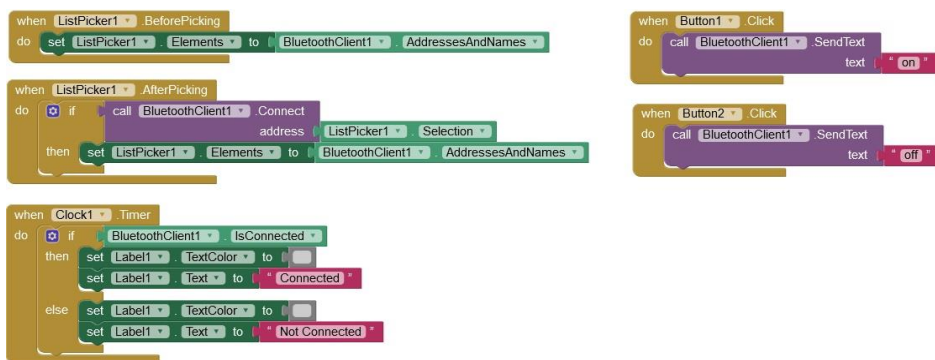
⁵ <https://appmaster.io>; <https://appmaster.io/pl/nocode-list/appy-pie>; <https://appmaster.io/pl/nocode-list/appsheet>; <https://appmaster.io/pl/nocode-list/glide>

dla początkujących powinny być łatwe w obsłudze, intuicyjne i niewymagające dużego nakładu pracy. Jednym z takich narzędzi jest MIT App Inventor.

Podczas projektowania aplikacji mobilnych ważne jest, aby uwzględnić różnice między platformami iOS i Android oraz przestrzegać ich konkretnych wytycznych projektowych. W tym kontekście decydujący dla metodyki nauczania jest wybór środowiska IDE i języków programowania. Literatura przedmiotu opisuje wiele różnych rozwiązań⁶.

Programowanie w środowisku MIT App Inventor

Ze względu na duże walory edukacyjne skoncentrowano się poniżej na bezpłatnym środowisku programistycznym MIT App Inventor przeznaczonym do tworzenia aplikacji na urządzenia mobilne pracujących pod systemami Android i iOS. MIT App Inventor to środowisko programistyczne umożliwiające tworzenie aplikacji mobilnych bezpośrednio w przeglądarce internetowej bez konieczności zaawansowanej znajomości programowania⁷. Środowisko oparte jest na metodzie „drag &drop”, co pozwala programistom tworzyć aplikacje poprzez łączenie bloków reprezentujących kod (rys. 1).



Rys. 1. Przykład programu w środowisku MIT App Inventor

Źródło: opracowanie własne.

MIT App Inventor oferuje łatwą komunikację z urządzeniami zewnętrznymi przy użyciu Bluetooth lub WiFi. Obsługuje mikrokontrolery Arduino, ESP, Raspberry Pi, umożliwiając tworzenie aplikacji sterujących robotami bądź systemami wbudowanymi w różnych urządzeniach mechatronicznych.

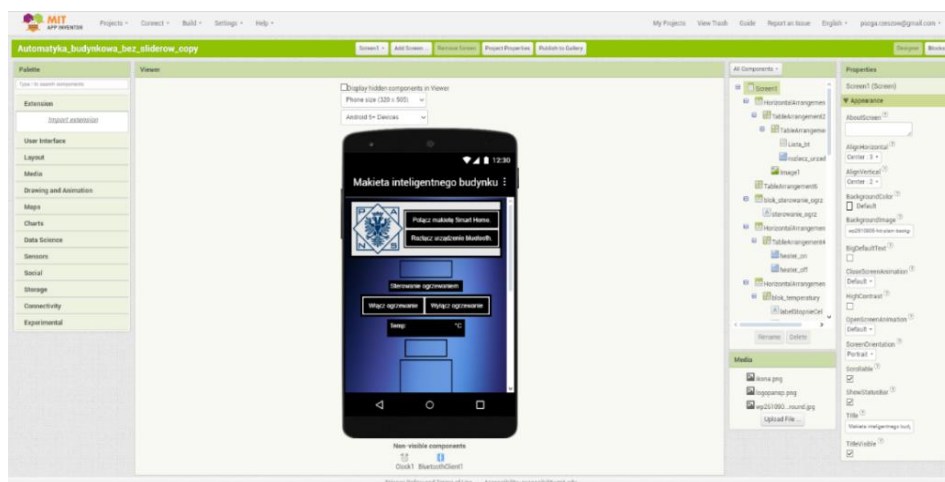
⁶ W. Jaśkowiec i in., *Aplikacje mobilne dla studenta i technika programisty*, Wyd. Itstart 2024; A. Stasiewicz, *Android. Podstawy tworzenia aplikacji*, Helion 2016.

⁷ <https://appinventor.mit.edu/>

Środowisko IDE zbudowane jest z dwóch części – Designer oraz Blocks. Część Designer odpowiada za cały wygląd aplikacji, który projektuje się przeciągając i opuszczając bloki funkcyjne, np. przyciski, suwaki, bloki Layout, pola tekstowe itp. Część Blocks odpowiada za logikę części Designer. Dzięki zastosowaniu modelu podobnego do języka Scratch programowanie odbywa się poprzez przeciąganie bloczków funkcyjnych, zmiennych oraz pętli⁸.

MIT APP Inventor umożliwia również symulację smartfona w czasie rzeczywistym, dzięki czemu możliwe jest przetestowanie aplikacji przed jej skompilowaniem. Wykorzystując moduł MIT AI2 Companion, użytkownik jest w stanie zasymulować na własnym smartfonie budowaną aplikację, testując jej działanie oraz rozłożenie elementów na ekranie.

Na rys. 2 przedstawiono środowisko MIT App Inventor podczas projektowania interfejsu aplikacji mobilnej do sterowania systemem automatyki budynkowej⁹. Graficzne IDE umożliwia łatwe rozmieszczenie elementów interfejsu, nadanie im identyfikatorów oraz zachowanie ich pozycji.



Rys. 2. Środowisko MIT App Inventor

Źródło: P. Ozga, *Projekt systemu automatyki budynkowej*, praca inżynierska INT PANS, Przemysł 2025.

W MIT App Inventor znajdują się narzędzia, które pozwalają początkującym studentom tworzyć oryginalne aplikacje AI, które jeszcze dekadę temu

⁸ F. Kamriani, K. Roy, *App Inventor 2 Essentials*, Packt Publishing 2016; K. Lang, S. Tezel, *Become an App Inventor: The Official Guide from MIT App Inventor*, Walker Books Ltd 2022.

⁹ P. Ozga, *Projekt systemu automatyki budynkowej*, praca inżynierska INT PANS, Przemysł 2025.

uchodziłyby za zaawansowane projekty badawcze¹⁰. Rozwiązania te otwierają przed uczniami nowe możliwości poznawania AI i wzmacniają ich pozycję jako twórców cyfrowej przyszłości. MIT App Inventor umożliwia eksportowanie gotowej aplikacji do formatu apk. W praktyce oznacza to, że można ją dystrybuować do użytkowników i opublikować w sklepie Google Play.

Projekt aplikacji mobilnej do sterowania robotem edukacyjnym

Aplikacje mobilne odgrywają kluczową rolę w zwiększaniu funkcjonalności i wygody użytkowania urządzeń mechatronicznych. Umożliwiają zdalne sterowanie, monitorowanie, programowanie oraz integrację z innymi systemami, co przyczynia się do rozwoju nowoczesnych technologii i poprawy jakości życia. Szczególnym obszarem systemów mechatronicznych jest robotyka. Aplikacje mobilne odgrywają podstawową rolę w robotyce, oferując możliwość zdalnego sterowania, programowania i monitorowania pracy robotów. Dzięki dynamicznemu rozwojowi technologii mobilnych smartfony i tablety stają się integralnym narzędziem w pracy z robotami, zarówno w edukacji, jak i przemyśle.

Sterowanie robotem mobilnym to kluczowy element jego funkcjonowania, umożliwiający przeprowadzanie złożonych zadań, takich jak poruszanie się, manipulowanie obiektami w przestrzeni czy sczytywanie danych z sensorów. Jednym z głównych zastosowań aplikacji mobilnych w robotyce jest więc zdalne sterowanie robotami. Przy użyciu takich technologii, jak Bluetooth, Wi-Fi czy NFC, aplikacje mobilne umożliwiają użytkownikom nawigację i kontrolowanie robotów w czasie rzeczywistym.

Na przykładzie robota edukacyjnego przedstawiono projektowanie aplikacji mobilnej¹¹, które oparto na metodyce Desing Thinking¹². Obecnie stanowi ona popularną metodę pracy projektowej stosowaną w wielu różnych dziedzinach na świecie. Zakłada innowacyjne podejście w celu rozwiązania problemów i wdrażania nowatorskich rozwiązań projektowych. W naszym przypadku metoda Desing Thinking umożliwiła tworzenie nowego robota poprzez dogłębną analizę celów kształcenia oraz określanie i rozwiązywanie problemów dywergencyjnych. Proces projektowania składał się z pięciu etapów, takich jak: empatyzacja, zdefiniowanie problemów, poszukiwanie różnych rozwiązań, prototypowanie i testowanie.

¹⁰ <https://appinventor.mit.edu/explore/ai-with-mit-app-inventor>

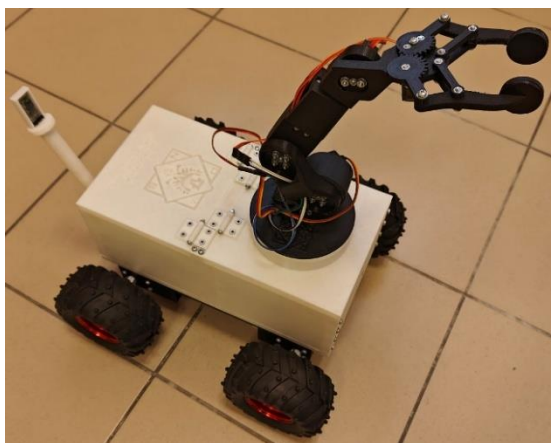
¹¹ W tej części opracowania wykorzystano informacje z pracy inżynierskiej przygotowanej pod kierunkiem D. Dutkowskiego: *Projekt robota mobilnego*, INT PANS, Przemysł 2025.

¹² N. Cross, *Desing Thinking – Understanding how designers think and work*, Wyd. Bloomsbury 2023.

W założeniach projektowych uwzględniono:

- komunikację przy pomocy połączenia Bluetooth z mobilną aplikacją sterującą zaprojektowaną w środowisku MIT App Inventor dla systemu Android;
- poruszanie się platformy w czterech kierunkach: przód, tył, obrót w lewo, obrót w prawo;
- ramię robotyczne z chwytakiem (ang. *gripper*) umożliwiające przenoszenie przedmiotów sterowane przez serwomechanizmy;
- czujniki kontrolujące orientację w terenie (rys. 3).

Projekt robota obejmował zarówno zaprogramowanie mikrokontrolera Arduino Mega, jak również aplikację mobilną.



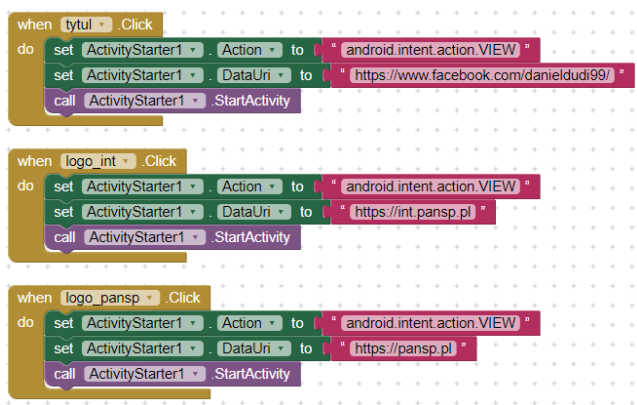
Rys. 3. Edukacyjny robot mobilny

Źródło: opracowanie własne.

Część logiczną aplikacji mobilnej podzielono na różne fragmenty. Prace projektowe rozpoczęto od zaprogramowania interfejsu użytkownika, który składa się z różnych modułów – wśród nich wyróżnia się część tytułową, wzbogaconą o funkcjonalność przekierowania. Po kliknięciu na poszczególne elementy, takie jak logo uczelni, tytuł aplikacji czy logo instytutu, użytkownik zostaje przekierowany na wybrane strony internetowe poprzez wywołanie funkcji `call.ActivityStarter1.StartActivity` (rys. 4).

Na rys. 5 przedstawiono interfejs użytkownika aplikacji mobilnej w dwóch stanach pracy – po lewej stronie bez nawiązanego połączenia z robotem, a po prawej w stanie połączenia przez Bluetooth. Część środkowa aplikacji służy do sterowania ruchem robota do przodu, do tyłu, obrót w lewo, obrót w prawo. W części tej znajdują się aktualne odczyty z czujników odległości. Odczyt górny przedstawia odległość z przodu od przeszkody, natomiast odczyt dolny z tyłu od

przeszkody. Część dolna interfejsu odpowiada za mobilność ramienia robotycznego i umożliwia sterowanie serwomechanizmami.



Rys. 4. Logika działania członu tytułowego

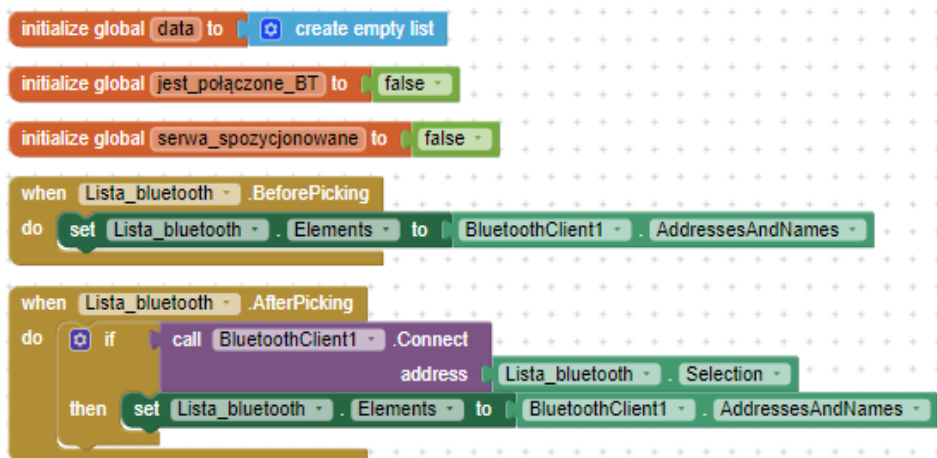
Źródło: opracowanie własne.



Rys. 5. Interfejs aplikacji mobilnej w dwóch stanach pracy

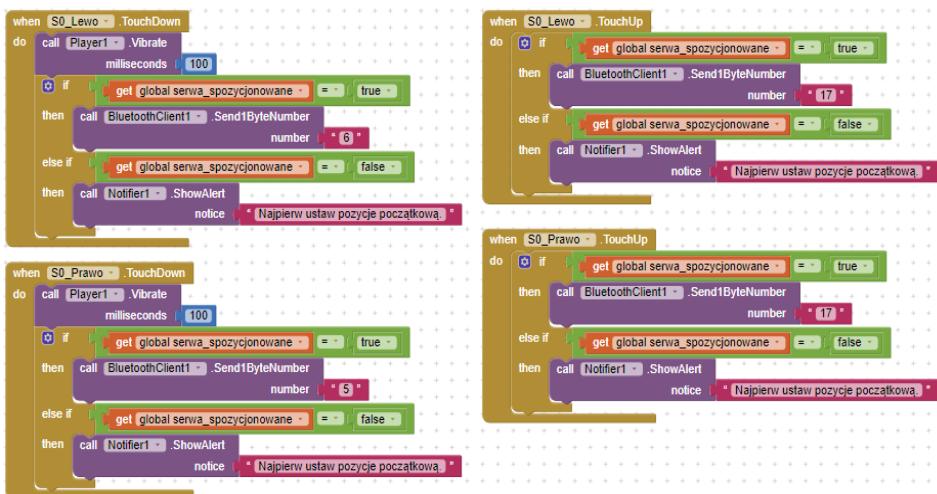
Źródło: D. Dutkowski, *Projekt robota mobilnego*, INT PANS, Przemysł 2025.

Na poniższych rysunkach przedstawiono wybrane fragmenty kodu aplikacji mobilnej odpowiedzialne za komunikację i sterowanie elementami robota. Na rys. 6 zaprezentowano inicjację łączności smartfona z robotem przez Bluetooth (tzw. parowanie), na rys. 7 zaś opisano sterowanie ramieniem robotycznym, czyli pracą serwomechanizmów.



Rys. 6. Logika bloku łączności

Źródło: opracowanie własne.



Rys. 7. Fragment logiki sterującej serwomechanizmami

Źródło: opracowanie własne.

Zakończenie

Celem niniejszego opracowania było zaprezentowanie praktycznych rozwiązań projektowania aplikacji mobilnych w edukacyjnych systemach mechatronicznych.

Poszukiwanie narzędzi do tworzenia aplikacji mobilnych dla początkujących programistów doprowadziło nas do MIT App Inventor. Zaprojektowany specjalnie dla edukacji MIT App Inventor proponuje konstruktywistyczny model uczenia się. Dzięki prostemu interfejsowi użytkownika wykorzystuje metodę „drag & drop” w procesie tworzenia aplikacji, wyjaśniając złożony świat kodowania początkującym programistom. MIT App Inventor wyróżnia nacisk na wartość edukacyjną. Platforma została stworzona, aby wspomóc nauczanie programowania i projektowania poprzez zastosowanie kodowania blokowego. To wizualne podejście okazuje się niezwykle intuicyjne, ponieważ użytkownicy łączą bloki kodu niczym elementy puzzli, aby stworzyć aplikację mobilną.

MIT App Inventor jest nie tylko narzędziem uczenia się programowania przez zabawę, ale potężnym środowiskiem IDE umożliwiającym tworzenie w pełni funkcjonalnych aplikacji na Androida i iOS. Użytkownicy mają dostęp do rozbudowanej biblioteki różnych bloków funkcjonalnych, a dużym atutem MIT App Inventor jest aktywna i zaangażowana społeczność. Dzieli się swoimi projektami, samouczkami i poradami z każdym, kto szuka rozwiązania problemów. Taka współpraca sprzyja indywidualnemu rozwojowi i zbiorowemu poszerzaniu wiedzy w danej dziedzinie i nie ogranicza użytkowników chcących tworzyć aplikacje wielopatformowe¹³. Otwarty charakter MIT App Inventor oznacza, że można odkrywać i modyfikować podstawy narzędzia, przyczyniając się do rozwoju platformy.

Podsumowując, programowanie aplikacji mobilnych znacząco wzbogaca edukację informatyczno-techniczną, czyniąc ją bardziej interaktywną, dostępną i angażującą intelektualnie uczniów. Umożliwia ono uczniom zdobywanie praktycznych umiejętności, realizację projektów DIY, a także dostęp do bogatych zasobów edukacyjnych. Programowanie sprawia, że nauka techniki staje się bardziej przystępna i atrakcyjna dla uczniów i studentów na różnych poziomach edukacji.

Bibliografia

- Cross N., *Desing Thinking – Understanding how designers think and work*, Wyd. Bloomsbury 2023.
Delessio C. i in., *Android Studio w 24 godziny. Wygodne programowanie dla platformy Android*, wyd. IV, Helion 2016.

¹³ Wielopatformowe aplikacje mobilne są kompatybilne z wieloma mobilnymi systemami operacyjnymi równocześnie, takimi jak iOS i Android.

Dutkowski D., *Projekt robota mobilnego*, praca inżynierska INT PANS, Przemysł 2025.
Jaśkowiec W. i in., *Aplikacje mobilne dla studenta i technika programisty*, Wyd. Itstart 2024.
Kamrioni F., Roy K., *App Inventor 2 Essentials*, Packt Publishing 2016.
Lang K., Tezel S., *Become an App Inventor: The Official Guide from MIT App Inventor*, Walker Books Ltd 2022.
Ozga P., *Projekt systemu automatyki budynkowej*, praca inżynierska INT PANS, Przemysł 2025.
Płonkowski M., *Android Studio. Tworzenie aplikacji mobilnych*, Helion 2017.
Stachowicz I., *Projektowanie aplikacji mobilnych – kluczowe zasady*, <https://innovatika.com/pl/projektowanie-aplikacji-mobilnych-kluczowe-zasady>, 2023.
Stasiewicz A., *Android. Podstawy tworzenia aplikacji*, Helion 2016.
Szymański G., *Projektowanie i reklama aplikacji mobilnych*, Wyd. Stowarzyszenie Innowatorów Zarządzania, Łódź 2021.

Netografia

<https://designthinking.pl/co-to-jest-design-thinking/>
<https://yarmobile.com/pl/projektowanie-aplikacji-mobilnych/>
<https://appmaster.io>
<https://appmaster.io/pl/nocode-list/appy-pi>
<https://appmaster.io/pl/nocode-list/appsheets>
<https://appmaster.io/pl/nocode-list/glide>
<https://appinventor.mit.edu/>
<https://appinventor.mit.edu/explore/ai-with-mit-app-inventor>