

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

KARTA OPISU PRZEDMIOTU

| | | | |
|------------------------|-----------|--|------------------|
| Wydział | | Informatyki | |
| Kierunek | | Informatyka | |
| Specjalność | | Programowanie | |
| Semestr | VI | Program studiów, dla którego obowiązuje sylabus | 2025/2026 |
| Stopień studiów | I | | |

| | | | | |
|----------------------|-------------------------------|------|----------------|----------------|
| Nazwa przedmiotu | Tworzenie aplikacji mobilnych | | | |
| Kod przedmiotu | ZSP | | | |
| Łączna liczba godzin | 30 | Tryb | stacjonarny | niestacjonarny |
| Profil kształcenia | Ogólnoakademicki (A) | | Praktyczny (P) | |
| Forma zajęć | laboratorium | | | |
| Język przedmiotu | polski | | | |
| Liczba punktów ECTS | 2 | | | |

| | |
|---------------------------------|---------------------|
| Prowadzący zajęcia | |
| Forma prowadzonych zajęć | Laboratorium |
| Wymiar zajęć | 30 h |
| Stopień (tytuł) naukowy | |
| Imię | |
| Nazwisko | |

| | |
|------------------------------------|--|
| Wymagania wstępne | Podstawowa znajomość języków programowania, takich jak Java lub C++. Umiejętność obsługi komputera i środowisk programistycznych. Znajomość podstawowych algorytmów i struktur danych. |
| Założenia i cele przedmiotu | Celem przedmiotu jest zapoznanie studentów z podstawami tworzenia aplikacji mobilnych na platformy iOS i Android oraz z podejściami natywnymi i cross-platformowymi. Studenci nauczą się korzystać z nowoczesnych frameworków mobilnych (np. React Native, Flutter), projektować czytelne i użyteczne interfejsy mobilne, integrować aplikacje z zewnętrznymi usługami oraz testować oprogramowanie na różnych urządzeniach. |
| Metody dydaktyczne | <ol style="list-style-type: none"> 1. Prezentacje multimedialne. 2. Pokazy przykładowych rozwiązań problemów. 3. Rozwiązywanie zadań praktycznych. |

| Efekty uczenia się (odniesienie do charakterystyk poziomów Polskiej Ramy Kwalifikacji) | | Odniesienie do efektów dla kierunku | Odniesienie do efektów uczenia się wg Polskiej Ramy Kwalifikacji |
|---|---|--|---|
| WIEDZA – absolwent zna i rozumie: | W01. Zasady tworzenia front-endu dla aplikacji mobilnych, w tym znaczenie responsywności, nawigacji i elementów interfejsu dostosowanych do ekranów dotykowych. | K_W02 K_W06 K_W10 | P6S_WG P6S_WG_INŻ |

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

| | | | |
|---|--|--|---|
| | <p>W02. Architektury mobilnych systemów operacyjnych oraz wzorce projektowe stosowane w tworzeniu aplikacji mobilnych.</p> <p>W03. Podstawowe paradygmaty programowania obiektowego i strukturalnego stosowane w środowiskach mobilnych.</p> <p>W04. Metody i narzędzia modelowania systemów mobilnych (np. UML), ułatwiające planowanie i dokumentację projektu aplikacji.</p> <p>W05. Podstawy korzystania z usług chmurowych i integracji aplikacji mobilnych z backendami w chmurze (np. Firebase, AWS Amplify).</p> | <p>K_W19</p> <p>K_W20</p> | |
| <p>UMIEJĘTNOŚCI</p> <p>– absolwent potrafi:</p> | <p>U01. Tworzyć proste aplikacje mobilne wykorzystując paradygmat obiektowy oraz wybrany język i środowisko deweloperskie (Java/Kotlin dla Android, Swift dla iOS).</p> <p>U02. Projektować i implementować responsywne interfejsy użytkownika mobilnych aplikacji, uwzględniając zasady UX/UI oraz dobre praktyki projektowe.</p> <p>U03. Integrować aplikacje mobilne z zewnętrznymi API i usługami, np. mapami, usługami autoryzacji czy analityki, korzystając z odpowiednich bibliotek i SDK.</p> <p>U04. Testować aplikacje mobilne, przeprowadzając testy funkcjonalne, wydajnościowe i użyteczności na symulatorach i urządzeniach fizycznych.</p> <p>U05. Projektować podstawową architekturę aplikacji mobilnej, korzystając z wzorców projektowych, zarządzania stanem oraz narzędzi automatyzacji.</p> | <p>K_U01</p> <p>K_U02</p> <p>K_U03</p> <p>K_U04</p> <p>K_U09</p> <p>K_U11</p> <p>K_U15</p> <p>K_U23</p> <p>K_U24</p> | <p>P6S_UW</p> <p>P6S_UW_INŻ</p> <p>P6S_UO</p> <p>P6S_KK</p> <p>P6S_UK</p> |
| <p>KOMPETENCJE SPOŁECZNE</p> <p>– absolwent jest gotów do</p> | <p>K01. Pracy w zespole, przyjmując w nim różne role.</p> <p>K02. Krytycznej oceny możliwości urządzeń oprogramowania i systemów dostępnych na rynku IT.</p> <p>K03. Ciągłego samokształcenia się w celu dostosowywania się do dynamicznie zmieniających się technologii.</p> | <p>K_K04</p> <p>K_K05</p> <p>K_K06</p> | <p>P6S_UO</p> <p>P6S_KR</p> <p>P6S_KK</p> |

| Lp. | Tematyka zajęć | Liczba godzin |
|-----------------------------------|--|---------------|
| Forma zajęć – laboratorium | | |
| 1 | Wprowadzenie do programowania mobilnego. Platformy (iOS, Android), natywne vs cross-platformowe rozwiązania. | 6 |

AKADEMIA TECHNICZNO-INFORMATYCZNA W NAUKACH STOSOWANYCH

| | | |
|---|--|---|
| 2 | Frameworki do tworzenia aplikacji mobilnych. React Native, Flutter, Xamarin. | 6 |
| 3 | Projektowanie interfejsów mobilnych. UX/UI dla aplikacji mobilnych, zasady projektowania interakcji. | 8 |
| 4 | Integracja z API i usługami mobilnymi. Praca z API, korzystanie z usług takich jak Firebase, Google Maps. | 6 |
| 5 | Testowanie aplikacji mobilnych. Narzędzia do testowania, debugowanie, testy na różnych urządzeniach. Zaliczenie. | 4 |

| | | |
|---|--|---|
| Forma i warunki zaliczenia przedmiotu | Wykonanie projektów. Częstkowe prezentacje, zdawanie raportów, obrona projektów. | |
| Metody weryfikacji efektów uczenia się | | Nr efektu uczenia się z sylabusu |
| | Ocena projektów i częstkowych prezentacji. | W01-W05, U01-U05, K01-K03 |

| | |
|---------------------------------|--|
| Literatura podstawowa | <ol style="list-style-type: none"> 1. R. C. Martin, <i>Czysty kod. Podręcznik dobrego programisty</i>, Helion, Gliwice 2010. 2. J. Roszkowski, <i>Analiza i projektowanie strukturalne</i>, Helion, Gliwice, 2004. 3. N. Wirth, <i>Algorytmy + struktury danych = programy</i>, WNT, Warszawa 2002. 4. A. Roman, <i>Testowanie i jakość oprogramowania. Modele, techniki, narzędzia</i>, Wydawnictwo Naukowe PWN, Warszawa 2015. |
| Literatura uzupełniająca | <ol style="list-style-type: none"> 1. S. Prata, <i>Język C++. Szkoła programowania</i>. Wydanie VI, Helion, Gliwice 2019. 2. B. Eckel, <i>Thinking in Java. Edycja polska</i>, Helion, Gliwice 2006. |

| Nakład pracy studenta | |
|---|---------------|
| | Liczba godzin |
| Zajęcia dydaktyczne | 30 |
| Przygotowanie się do zajęć | 5 |
| Studiowanie literatury | 5 |
| Udział w konsultacjach | 2 |
| Przygotowanie projektu / eseju / prezentacji itp. | 18 |
| Przygotowanie się do egzaminu / zaliczenia | - |
| Inne | - |
| ŁĄCZNY nakład pracy studenta w godz. | 60 |
| Liczba punktów ECTS | 2 |