

**Wymagania edukacyjne niezbędne do
otrzymania przez ucznia śródrocznej i rocznej
oceny klasyfikacyjnej z pracowni
Programowania aplikacji mobilnych
Klasa 3 technikum**

Wymagania edukacyjne niezbędne do otrzymania przez ucznia śródrocznej oceny klasyfikacyjnej

temat lekcji	zagadnienia	Wymagania edukacyjne na poszczególne oceny				
		ocena dopuszczająca uczeń	ocena dostateczna uczeń	ocena dobra uczeń	ocena bardzo dobra uczeń	ocena celująca uczeń
Rozdziału 1: Przygotowanie środowiska pracy:						
Wybór odpowiedniego IDE dla aplikacji mobilnych	IDE, wybór, aplikacje mobilne, platformy, środowisko pracy.	Zrozumienie pojęcia IDE. Wymienienie co najmniej jednego IDE dedykowanego dla aplikacji mobilnych.	Zrozumienie różnic między co najmniej dwoma popularnymi IDE dla aplikacji mobilnych. Wybór odpowiedniego IDE w zależności od wymagań projektu.	Analiza funkcjonalności, zalet i wad co najmniej trzech IDE dla aplikacji mobilnych. Zrozumienie integracji IDE z innymi narzędziami.	Biegłość w korzystaniu z wybranego IDE, w tym skorzystanie z zaawansowanych funkcji. Porównanie możliwości rozszerzenia funkcjonalności IDE za pomocą pluginów i dodatków.	Biegłość w pracy z różnymi IDE, zdolność szybkiego dostosowywania się do nowego środowiska. Analiza i rozwiązanie skomplikowanych problemów związanych z IDE.
Konfiguracja środowiska programistycznego	Konfiguracja, środowisko programistyczne, ustawienia, narzędzia, wtyczki.	Zrozumienie podstaw konfiguracji IDE. Zastosowanie podstawowych ustawień dla wybranego IDE.	Znajomość narzędzi i opcji dostępnych w IDE do konfiguracji środowiska. Zastosowanie odpowiedniej konfiguracji w zależności od wymagań projektu.	Skonfigurowanie środowiska pod kątem specyficznych potrzeb projektu. Zrozumienie zaawansowanych opcji konfiguracji w	Optymalizacja środowiska programistycznego dla efektywnej pracy. Integracja IDE z zewnętrznymi narzędziami i usługami.	Biegłość w dostosowywaniu środowiska programistycznego o pod różne potrzeby i technologie.

				wybranych IDE.		
Podstawy składni i środowiska dla popularnych języków programowania mobilnego (np. Java dla Androida, Swift dla iOS)	Składnia, Java, Android, Swift, iOS, podstawy, środowisko.	Zrozumienie podstawowej składni jednego z popularnych języków programowania mobilnego. Uruchomienie prostej aplikacji w wybranym języku.	Zastosowanie podstawowych konstrukcji językowych w praktyce. Znajomość środowiska wybranego języka (np. SDK dla Androida).	Pisanie bardziej zaawansowanych aplikacji w wybranym języku. Znajomość narzędzi i bibliotek dedykowanych dla danego języka.	Implementacja skomplikowanych funkcjonalności w aplikacji mobilnej, korzystając z zaawansowanych aspektów języka. Zrozumienie i zastosowanie najlepszych praktyk w programowaniu mobilnym.	Biegłość w programowaniu w co najmniej dwóch popularnych językach programowania mobilnego.
rozdział 2 Zasady programowania aplikacji mobilnych						
Cykl życia aplikacji mobilnej	Cykl życia, fazy, inicjalizacja, zakończenie, aplikacje mobilne.	Zrozumienie podstawowych etapów cyklu życia aplikacji mobilnej.	Znajomość różnic w cyklach życia aplikacji na różnych platformach (np. Android vs. iOS).	Zastosowanie wiedzy o cyklu życia podczas tworzenia aplikacji mobilnej.	Zaawansowana znajomość i umiejętność zarządzania cyklem życia aplikacji w praktycznych projektach.	Optymalizacja wydajności aplikacji przez skuteczne zarządzanie cyklem życia.
Programowanie reaktywne vs. programowanie imperatywne w aplikacjach mobilnych	Programowanie reaktywne, imperatywne, różnice, zalety, wady.	Zrozumienie podstawowych różnic między programowaniem reaktywnym a imperatywnym.	Umiejętność podania przykładów zastosowań programowania reaktywnego i imperatywnego w kontekście mobilnym.	Zastosowanie w praktyce zarówno programowania reaktywnego, jak i imperatywnego w projektach mobilnych.	Zaawansowana znajomość i efektywne korzystanie z technik programowania reaktywnego w zaawansowanych projektach	Optymalizacja aplikacji mobilnych poprzez wybór odpowiedniego podejścia programistycznego w zależności od problemu
Zarządzanie pamięcią	Pamięć, zarządzanie,	Zrozumienie	Identyfikacja	Zastosowanie	Efektywne	Zaawansowane

i zasobami w aplikacjach mobilnych	zasoby, optymalizacja, zarządzanie pamięcią.	podstaw zarządzania pamięcią w aplikacjach mobilnych.	potencjalnych problemów związanych z zarządzaniem pamięcią w aplikacjach mobilnych.	podstawowych technik zarządzania pamięcią w praktyce.	zarządzanie zasobami, w tym optymalizacja zużycia pamięci i baterii.	techniki zarządzania pamięcią, w tym diagnostyka i rozwiązywanie skomplikowanych problemów związanych z zarządzaniem zasobami.
rozdział 3 Projektowanie aplikacji						
Badania UX i ich rola w projektowaniu aplikacji	Badania, UX, projektowanie, interakcje, użytkownik.	Zrozumienie podstawowych pojęć związanych z UX. Wiedza na temat znaczenia badań UX w projektowaniu aplikacji.	Zastosowanie podstawowych technik badań UX w praktyce. Interpretacja wyników prostych badań UX.	Projektowanie interfejsu opartego na wynikach badań UX. Zastosowanie zaawansowanych technik badań UX.	Biegłość w korzystaniu z różnych metod badawczych. Analiza i implementacja wyników badań UX w trudniejszych projektach.	Innowacyjne wykorzystanie technik UX w projektowaniu aplikacji. Krytyczna analiza i ocena różnych metod badawczych w kontekście UX.
Narzędzia do tworzenia prototypów i mockupów interfejsu	Narzędzia, prototypy, mockupy, projektowanie, wizualizacja.	Znajomość podstawowych narzędzi do tworzenia prototypów. Stworzenie prostego mockupu aplikacji.	Umiejętność korzystania z zaawansowanych funkcji wybranego narzędzia do prototypowania.	Tworzenie interaktywnych prototypów. Zastosowanie narzędzi do testowania prototypów z użytkownikami.	Zaawansowane techniki prototypowania i testowania. Zastosowanie narzędzi prototypowania w złożonych projektach.	Optymalizacja procesu projektowania opartego na prototypach. Krytyczna ocena i analiza różnych narzędzi i technik prototypowania.
Integracja z zewnętrznymi	Integracja, API, zewnętrzne usługi,	Zrozumienie, czym jest API i	Integracja prostej aplikacji mobilnej z	Korzystanie z różnorodnych	Zaawansowane techniki integracji z	Biegłość w integracji

usługami i API	dane, interfejs.	jakie jest jego zastosowanie w aplikacjach mobilnych.	jednym zewnętrznym API.	zewnętrznych usług w aplikacji. Znajomość bezpieczeństwa podczas integracji z API.	wieloma różnorodnymi usługami zewnętrznymi. Optymalizacja komunikacji z zewnętrznymi usługami.	skomplikowanych, wielopoziomowych usług zewnętrznych.
rozdział 4 Architektoniczne projektowanie aplikacji						
Podział aplikacji na komponenty i moduły	Podział, komponenty, moduły, architektura, struktura.	Zrozumienie podstawowych pojęć: komponent i moduł w kontekście architektury oprogramowania.	Umiejętność podziału funkcjonalności aplikacji na logiczne komponenty.	Zaawansowany podział aplikacji na komponenty z uwzględnieniem zasad SOLID. Zastosowanie zależności między komponentami i modułami w sposób efektywny.	Optymalizacja struktury aplikacji przez skuteczny podział na komponenty i moduły.	Projektowanie zaawansowanych systemów złożonych z wielu powiązanych komponentów i modułów.
MVP, MVVM i inne architektury dla aplikacji mobilnych	MVP, MVVM, wzorce architektoniczne, aplikacje mobilne, struktura.	Zrozumienie podstawowych pojęć architektur MVP i MVVM.	Zastosowanie jednej z architektur (MVP lub MVVM) w prostym projekcie aplikacji mobilnej.	Biegłość w projektowaniu aplikacji zarówno w architekturze MVP, jak i MVVM. Zrozumienie zalet i wad obu podejść.	Analiza i zastosowanie innych architektur w projektowaniu aplikacji mobilnych. Optymalizacja wydajności aplikacji poprzez wybór odpowiedniej architektury.	Zaawansowane techniki projektowania w różnych architekturach. Adaptacja i łączenie różnych architektur w jednym projekcie w celu osiągnięcia optymalnych

						rozwiązań.
Zarządzanie stanem w aplikacjach mobilnych	Stan, zarządzanie, aplikacje mobilne, danych, sesja.	Zrozumienie pojęcia "stan" w kontekście aplikacji mobilnych.	Podstawowe techniki zarządzania stanem w aplikacji.	Zastosowanie narzędzi i bibliotek do efektywnego zarządzania stanem w aplikacji mobilnej.	Zaawansowane strategie zarządzania stanem w złożonych aplikacjach.	Optimalizacja wydajności i doświadczenia użytkownika poprzez biegłe zarządzanie stanem aplikacji.
rozdział 5 Testowanie oprogramowania						
Automatyczne testy jednostkowe w aplikacjach mobilnych	Automatyzacja, testy jednostkowe, testowanie, kod.	Zrozumienie pojęcia testu jednostkowego. Napisanie prostego testu jednostkowego dla funkcji lub metody w aplikacji.	Zastosowanie testów jednostkowych do większego fragmentu kodu.	Zastosowanie zaawansowanych technik testowania, takich jak mockowanie czy iniekcja zależności w testach jednostkowych.	Prowadzenie zaawansowanego testowania jednostkowego dla większych i bardziej złożonych aplikacji.	Pełne pokrycie kodu testami jednostkowymi. Biegłość w wykorzystaniu różnych technik i narzędzi do testowania jednostkowego.
Testy integracyjne i end-to-end w środowisku mobilnym	Testy integracyjne, end-to-end, scenariusze, interakcje.	Zrozumienie różnicy między testami jednostkowymi, integracyjnymi i end-to-end.	Przeprowadzenie podstawowych testów integracyjnych dla kluczowych elementów aplikacji.	Korzystanie z narzędzi do automatycznego testowania integracyjnego. Wykrywanie i naprawa błędów zidentyfikowanych podczas testów integracyjnych.	Zastosowanie testów end-to-end w środowisku mobilnym. Optimalizacja procesów w aplikacji na podstawie wyników testów end-to-end.	Zaawansowane strategie testowania integracyjnego i end-to-end.

Narzędzia do testowania i ich konfiguracja	Narzędzia, testowanie, konfiguracja, środowisko, testy.	Zrozumienie znaczenia narzędzi w procesie testowania. Znajomość co najmniej jednego narzędzia do testowania.	Konfiguracja wybranego narzędzia testującego dla konkretnego projektu. Zastosowanie narzędzia do przeprowadzenia podstawowych testów.	Zaawansowana konfiguracja i dostosowywanie narzędzi testujących do indywidualnych potrzeb projektu.	Zintegrowanie narzędzi testujących z innymi elementami środowiska deweloperskiego (np. systemami kontroli wersji, CI/CD).	Ekspertyza w wykorzystaniu i konfiguracji zaawansowanych narzędzi testujących. Optymalizacja procesów testowania poprzez efektywne wykorzystanie narzędzi.
rozdział 6 Tworzenie dokumentacji testowej						
Przykłady dokumentacji testów dla aplikacji mobilnych	Dokumentacja, testy, przykłady, specyfikacje.	Zrozumienie znaczenia dokumentacji testowej. Znajomość podstawowych elementów dokumentacji testowej.	Stworzenie prostego dokumentu opisującego testy dla wybranego elementu aplikacji. Analiza przykładowych dokumentów testowych i wyciągnięcie wniosków.	Tworzenie kompleksowej dokumentacji testowej dla większych fragmentów aplikacji. Umiejętność analizy i krytyki istniejących dokumentów testowych.	Zaawansowane techniki tworzenia dokumentacji, w tym wykorzystanie narzędzi i standardów branżowych.	Biegłość w tworzeniu, aktualizowaniu i zarządzaniu kompleksową dokumentacją testową dla dużych projektów.
Znaczenie i rola dokumentacji w cyklu życia oprogramowania	Znaczenie, rola, dokumentacja, cykl życia, oprogramowanie.	Zrozumienie podstawowej roli dokumentacji w procesie tworzenia oprogramowania.	Rozróżnienie różnych typów dokumentacji i ich roli w różnych etapach cyklu życia oprogramowania.	Analiza wpływu dobrej dokumentacji na jakość i efektywność procesu tworzenia oprogramowania	Zastosowanie wiedzy o roli dokumentacji w praktycznych projektach, uwzględniając zarówno korzyści, jak i wyzwania z	Zaawansowane strategie wykorzystania dokumentacji w celu optymalizacji pracy zespołu, komunikacji z interesariuszami i

				a.	nią związane.	zarządzania projektem.
Wykorzystanie narzędzi do tworzenia i zarządzania dokumentacją testową	Narzędzia, dokumentacja, generowanie, zarządzanie.	Znajomość podstawowych narzędzi służących do tworzenia dokumentacji.	Zastosowanie wybranego narzędzia do stworzenia podstawowej dokumentacji testowej.	Wykorzystanie zaawansowanych funkcji narzędzi do tworzenia dokumentacji. Integracja narzędzia do dokumentacji z innymi narzędziami używanymi w projekcie.	Zaawansowane strategie zarządzania dokumentacją testową za pomocą specjalistycznych narzędzi.	Zastosowanie automatycznych procesów (np. automatyczna generacja dokumentacji) w celu usprawnienia zarządzania dokumentacją testową.

Wymagania edukacyjne niezbędne do otrzymania przez ucznia rocznej oceny klasyfikacyjnej (obejmują wymagania edukacyjne niezbędne do otrzymania przez ucznia śródrocznej oceny klasyfikacyjnej).

rozdział 7 Metodologie prowadzenia projektu informatycznego

Agile vs. Waterfall w projektach aplikacji mobilnych	Agile, Waterfall, metodologie, projektowanie, rozwój.	Zrozumienie podstawowych różnic między metodologiami Agile i Waterfall. Znajomość etapów i cech charakterystycznych dla każdej z metodologii.	Umiejętność wyjaśnienia zalet i wad obu podejść w kontekście projektów aplikacji mobilnych.	Aktywne uczestniczenie w projekcie opartym na jednej z metodologii, z rozumieniem jej założeń.	Zaawansowane zastosowanie technik Agile lub Waterfall w trudniejszych projektach. Optymalizacja procesów projektowych na podstawie wybranego podejścia.	Ekspertyza w dostosowywaniu i łączeniu różnych metodologii w jednym projekcie. Prowadzenie szkoleń i warsztatów na temat wybranej metodologii.
Role w zespołach projektowych (Scrum	Role, zespoły, Scrum Master, Product Owner,	Zrozumienie podstawowych ról	Zastosowanie w praktyce wiedzy o	Aktywna współpraca z	Zaawansowane zrozumienie	Liderowanie zespołu

Master, Product Owner itp.)	zespół.	w zespołach projektowych i ich odpowiedzialność i.	roli, którą się pełni (np. działanie jako członek zespołu Scrum).	innymi rolami w zespole, rozumienie ich funkcji i oczekiwań. Umiejętność efektywnego komunikowania się z różnymi członkami zespołu.	dynamiki zespołowej i interakcji między różnymi rolami w zespole.	projektowego, mentorowanie młodszych członków zespołu i optymalizacja pracy zespołu poprzez efektywne zarządzanie rolami.
Narzędzia do zarządzania projektem i śledzenia postępów	Narzędzia, zarządzanie, projekty, postępy, planowanie.	Znajomość podstawowych narzędzi do zarządzania projektem, takich jak listy zadań lub prosty tracker.	Zastosowanie narzędzia do zarządzania projektem w praktycznym projekcie.	Zaawansowane wykorzystanie narzędzi do zarządzania projektem, takich jak JIRA, Trello itp.	Optymalizacja procesów zarządzania projektem poprzez skuteczne wykorzystanie narzędzi.	Ekspertyza w wyborze, konfiguracji i wykorzystywaniu zaawansowanych narzędzi do zarządzania projektem.
rozdział 8 Programowanie w C#						
Podstawy języka C#	C#, podstawy, język programowania, .NET, składnia.	Zrozumienie składni podstawowych instrukcji w C# (zmiennne, pętle, instrukcje warunkowe).	Zastosowanie kolekcji, takich jak listy i tablice. Wykorzystanie podstawowych klas i metod z biblioteki .NET.	Zrozumienie i zastosowanie podstawowych technik programowania obiektowego w C# (klasy, obiekty, hermetyzacja).	Efektywne zarządzanie pamięcią, zrozumienie mechanizmu garbage collection. Zastosowanie zaawansowanych funkcji języka, takich jak delegaty i wyrażenia lambda.	Biegłość w pisaniu skomplikowanego kodu w C#, optymalizacja wydajności i refaktoryzacja.
Programowanie w C#	C#, aplikacje mobilne,	Znajomość	Tworzenie prostych	Integracja	Zaawansowane	Biegłość w

dla aplikacji mobilnych	programowanie, rozwiązanie.	podstawowych bibliotek i narzędzi używanych w programowaniu aplikacji mobilnych w C#.	aplikacji mobilnych w C# z wykorzystaniem odpowiednich frameworków (np. Xamarin).	aplikacji mobilnej z zewnętrznymi usługami i API. Optymalizacja aplikacji mobilnych pod kątem wydajności i zużycia zasobów.	techniki programowania aplikacji mobilnych w C#, takie jak tworzenie animacji, zarządzanie stanem aplikacji i interakcja z urządzeniem.	tworzeniu zaawansowanych aplikacji mobilnych w C#, które są skalowalne, wydajne i łatwe w utrzymaniu.
Tworzenie klas i obiektów w C#	Klasy, obiekty, programowanie obiektowe, C#, metody.	Zrozumienie podstawowych pojęć związanych z programowaniem obiektywnym: klasa, obiekt, metoda, właściwość.	Tworzenie prostych klas i instancji obiektów, wykorzystanie konstruktorów.	Zastosowanie dziedziczenia i kompozycji w praktyce. Używanie modyfikatorów dostępu do zarządzania widocznością składników klasy.	Zastosowanie zaawansowanych technik programowania obiektowego, takich jak interfejsy, abstrakcja i polimorfizm.	Biegłość w tworzeniu skomplikowanych hierarchii klas, projektowanie efektywnych i optymalnych struktur obiektowych.
Dziedziczenie i polimorfizm w C#	Dziedziczenie, polimorfizm, klasy, interfejsy, C#.	Zrozumienie podstaw dziedziczenia i polimorfizmu w kontekście języka C#.	Stosowanie dziedziczenia w praktyce, rozumienie relacji między klasami bazowymi a pochodnymi.	Aktywne wykorzystanie polimorfizmu w projektach, zarówno w przypadku polimorfizmu nadpisywania, jak i polimorfizmu parametrycznego.	Zaawansowane techniki dziedziczenia i polimorfizmu, takie jak klasy abstrakcyjne, interfejsy i generyki.	Optymalizacja i refaktoryzacja

Obsługa wyjątków w C#	Wyjątki, obsługa, błędy, try-catch, C#.	Zrozumienie podstawowego pojęcia wyjątku w kontekście programowania w C#. Umiejętność wykrywania i reagowania na prosty wyjątek używając bloków try-catch.	Wykorzystanie bloku finally do zarządzania zasobami niezależnie od wystąpienia wyjątku.	Zastosowanie zaawansowanych technik obsługi wyjątków, takich jak tworzenie własnych klas wyjątków.	Efektywne zarządzanie wyjątkami w skomplikowanych scenariuszach, takich jak wielowątkowość czy asynchroniczność.	Ekspertyza w obsłudze wyjątków, w tym propagacja, logowanie i reagowanie na wyjątki w zaawansowanych architekturach aplikacji.
Praca z plikami w C#	Pliki, operacje, I/O, C#, strumienie.	Znajomość podstawowych operacji na plikach: odczyt, zapis, tworzenie i usuwanie. Używanie klas takich jak File i StreamReader do prostych operacji na plikach.	Obsługa różnych formatów plików, takich jak tekstowe, binarne czy XML. Zarządzanie ścieżkami i katalogami przy użyciu klasy Directory.	Zaawansowana praca z plikami, w tym operacje asynchroniczne i buforowane operacje wejścia/wyjścia.	Zastosowanie streamów do efektywnej pracy z dużymi plikami lub strumieniami danych.	Ekspertyza w zarządzaniu danymi przy użyciu plików, w tym optymalizacja, bezpieczeństwo i wydajność operacji wejścia/wyjścia.
rozdział 9 Visual Studio						
Środowisko programistyczne Visual Studio dla aplikacji mobilnych	Visual Studio, środowisko pracy, IDE, Microsoft. tworzenie, konfiguracja, aplikacje.Microsoft.	Zrozumienie podstawowych funkcji środowiska Visual Studio. Uruchamianie i edytowanie prostego projektu w Visual Studio.	Konfiguracja środowiska Visual Studio dla konkretnych potrzeb projektu aplikacji mobilnej. Znajomość podstawowych narzędzi i zakładki	Optymalizacja środowiska pracy w Visual Studio poprzez dostosowywanie ustawień, wtyczek i narzędzi	Zaawansowane techniki pracy w Visual Studio, takie jak profilowanie kodu, analiza statyczna i wykorzystanie narzędzi do debugowania.	Ekspertyza w konfiguracji i dostosowywaniu Visual Studio do różnorodnych, zaawansowanych projektów aplikacji mobilnych.

			dostępnych w Visual Studio.			
Tworzenie projektów w Visual Studio dla aplikacji mobilnych	Projekty, Visual Studio, tworzenie, konfiguracja, aplikacje.	Stworzenie prostego projektu aplikacji mobilnej w Visual Studio. Zrozumienie struktury projektu i plików w nim zawartych.	Zastosowanie szablonów projektowych dostępnych w Visual Studio dla różnych typów aplikacji mobilnych.	Zaawansowane techniki pracy z projektem, takie jak integracja z systemami kontroli wersji czy automatyzacja procesów budowy.	Optimalizacja i refaktoryzacja projektu w Visual Studio, wykorzystując dostępne narzędzia.	Biegłość w zarządzaniu skomplikowanymi projektami wielomodułowymi w Visual Studio.
Debugowanie kodu w Visual Studio dla aplikacji mobilnych	Debugowanie, Visual Studio, narzędzia, błędy, kod.	Zrozumienie podstawowych technik debugowania w Visual Studio. Użycie podstawowych narzędzi, takich jak punkty przerwania czy konsola wyjścia.	Śledzenie zmiennych i stosu podczas debugowania. Wykorzystanie narzędzi do analizy wyjątków.	Zaawansowane techniki debugowania, takie jak debugowanie warunkowe, punkty przerwania i śledzenie wydajności.	Optimalizacja kodu na podstawie informacji uzyskanych podczas debugowania. Użycie narzędzi do analizy statycznej i dynamicznej w celu identyfikacji problemów.	Ekspertyza w debugowaniu skomplikowanych problemów, integracja zewnętrznych narzędzi do debugowania w Visual Studio.
Zarządzanie projektem w Visual Studio dla aplikacji mobilnych	Zarządzanie, projekty, Visual Studio, zespoły, kontrola wersji.	Zrozumienie podstawowego przepływu pracy w Visual Studio. Zarządzanie prostym projektem w środowisku.	Organizacja i strukturacja kodu źródłowego za pomocą odpowiednich narzędzi w Visual Studio.	Zaawansowane techniki zarządzania projektem, takie jak integracja z CI/CD, testy automatyczne w Visual Studio.	Optimalizacja i refaktoryzacja kodu za pomocą wbudowanych narzędzi w Visual Studio. Zaawansowane strategie zarządzania wersjami i	Ekspertyza w zarządzaniu projektami w Visual Studio, prowadzenie zaawansowanych analiz kodu i optymalizacja procesów pracy w zespole.

					współpracy zespołowej w środowisku.	
Wdrażanie aplikacji na urządzenia mobilne z Visual Studio	Wdrażanie, Visual Studio, aplikacje mobilne, publikacja, dystrybucja.	Zrozumienie podstawowego procesu wdrażania aplikacji na urządzenia mobilne. Umiejętność zbudowania i wdrożenia prostej aplikacji na emulatorze.	Wdrażanie aplikacji na rzeczywiste urządzenia z różnymi konfiguracjami i systemami operacyjnymi.	Optymalizacja aplikacji pod kątem różnych urządzeń i rozdzielczości. Zarządzanie wersjami aplikacji i ich aktualizacjami.	Zaawansowane techniki wdrażania, takie jak automatyzacja procesów wdrażania, integracja z chmurą i usługami backendowymi	Ekspertyza w wdrażaniu skomplikowanych aplikacji mobilnych z Visual Studio, z uwzględnieniem zabezpieczeń, wydajności i skalowalności.
Projektowanie interfejsu użytkownika w XAML	XAML, interfejs użytkownika, projektowanie, kontrolki, Visual Studio.	Podstawowa znajomość składni XAML. Znajomość podstawowych kontrolki i ich atrybutów w XAML.	Projektowanie prostych interfejsów użytkownika z wykorzystaniem kontrolerów i stylów w XAML	Zaawansowane techniki projektowania w XAML, takie jak tworzenie niestandardowych kontrolerów i animacji.	Integracja XAML z logiką aplikacji, umiejętność tworzenia złożonych, interaktywnych interfejsów.	Ekspertyza w projektowaniu interfejsu użytkownika w XAML. Optymalizacja UX/UI, prowadzenie testów A/B i dostosowywanie interfejsu do potrzeb użytkowników.
rozdział 10 Podstawy języka XAML						
Projektowanie interfejsu użytkownika w XAML dla aplikacji mobilnych	XAML, projektowanie, interfejs użytkownika, kontrolki, layout.	Zrozumienie podstaw składni XAML. Umiejętność	Zastosowanie rozmieszczenia kontrolki w różnych typach	Wykorzystanie związania danych (data binding) do	Optymalizacja interfejsu użytkownika w XAML pod kątem	Projektowanie zaawansowanych, spersonalizowanych interfejsów

		stworzenia prostego interfejsu użytkownika z wykorzystaniem standardowych kontrolki XAML.	paneli (np. StackPanel, Grid). Stosowanie stylów i tematów w XAML do jednolitego wyglądu aplikacji.	prezentacji dynamicznych danych w interfejsie.	różnych rozdzielczości ekranu i orientacji urządzenia.	użytkownika, które zapewniają doskonałe doświadczenie użytkownika.
Tworzenie niestandardowych kontrolerów interfejsu użytkownika w XAML dla aplikacji mobilnych	Kontrolki, XAML, niestandardowe, projektowanie, komponenty.	Zrozumienie konceptu niestandardowych kontrolki. Tworzenie prostych niestandardowych kontrolki, bazując na istniejących kontrolkach.	Wykorzystanie właściwości i zdarzeń w niestandardowych kontrolkach. Stylizacja i adaptacja niestandardowych kontrolki do różnych środowisk.	Zaawansowane techniki tworzenia niestandardowych kontrolki, takie jak kompozycja, dziedziczenie i przesłanianie.	Optymalizacja wydajności niestandardowych kontrolki i ich dostosowywanie do różnych sytuacji użycia.	Biegłość w tworzeniu skomplikowanych, wielozadaniowych niestandardowych kontrolki, które są skalowalne i łatwe w utrzymaniu.
Animacje i efekty specjalne w XAML dla aplikacji mobilnych	Animacje, efekty, XAML, interakcje, UI.	Zrozumienie podstawowych mechanizmów animacji w XAML. Stworzenie prostych animacji właściwości.	Wykorzystanie kluczowych klasek w animacjach. Stosowanie przejść i transformacji w interfejsie użytkownika.	Zastosowanie zaawansowanych technik animacji, takich jak animacje scenariuszowe i grupowe.	Optymalizacja animacji pod kątem wydajności i płynności działania na urządzeniach mobilnych.	Tworzenie zaawansowanych, interaktywnych efektów specjalnych, które wzbogacają doświadczenie użytkownika i uczynią aplikację bardziej angażującą.
Tworzenie szablonów interfejsu użytkownika w XAML dla aplikacji mobilnych	Szablony, XAML, stylizacja, kontrolki, UI.	Zrozumienie podstawowego pojęcia szablonów w	Dostosowywanie i modyfikowanie szablonów kontrolki.	Zaawansowane techniki pracy z szablonami, takie jak	Optymalizacja szablonów pod kątem wydajności i użyteczności w	Ekspertyza w projektowaniu i implementacji skomplikowanych

		XAML. Stosowanie podstawowych szablonów do kontrolek.	Wykorzystanie szablonów danych do prezentacji zbiorów danych.	szablony hierarchiczne i selektory szablonów.	różnych scenariuszach.	szablonów, które są uniwersalne, elastyczne i dostosowane do różnorodnych potrzeb użytkowników.
rozdział 11 Xamarin						
Wprowadzenie do Xamarin dla aplikacji mobilnych	Xamarin, wieloplatformowość, aplikacje mobilne, .NET, Visual Studio.	Zrozumienie podstawowego celu i funkcji Xamarin jako platformy do tworzenia aplikacji wieloplatformowych. Instalacja i konfiguracja Xamarin w środowisku Visual Studio.	Umiejętność stworzenia prostego projektu Xamarin.Forms i jego uruchomienie na emulatorze.	Zrozumienie zaawansowanych funkcji Xamarin, takich jak wstrzykiwanie zależności i dostęp do platformowych API.	Zaawansowane techniki pracy z Xamarin, takie jak optymalizacja wydajności i personalizacja UI dla różnych platform.	Ekspertyza w korzystaniu z Xamarin do tworzenia rozbudowanych, wydajnych i skalowalnych aplikacji wieloplatformowych.
Tworzenie wieloplatformowych aplikacji mobilnych w C# i Visual Studio z Xamarin.Forms	Xamarin.Forms, aplikacje wieloplatformowe, C#, Visual Studio.	Zrozumienie podstawowego przepływu tworzenia aplikacji w Xamarin.Forms. Umiejętność tworzenia prostego interfejsu użytkownika z wykorzystaniem	Wykorzystanie rozmieszczenia kontrolek w różnych typach układów w Xamarin.Forms. Stosowanie stylów i tematów w Xamarin.Forms do jednolitego wyglądu aplikacji na różnych	Zaawansowane techniki pracy z Xamarin.Forms, takie jak tworzenie niestandardowych rendererów i korzystanie z platformowych rozszerzeń.	Optymalizacja aplikacji Xamarin.Forms pod względem wydajności i doświadczenia użytkownika.	Ekspertyza w tworzeniu rozbudowanych aplikacji wieloplatformowych w Xamarin.Forms, uwzględniając najnowsze trendy i technologie w branży mobilnej.

		standardowych kontrolerek Xamarin.Forms.	platformach.			
Tworzenie natywnych interfejsów użytkownika dla Androida, iOS i Windows Phone z Xamarin.Android, Xamarin.iOS i Xamarin.Forms	Xamarin.Android, Xamarin.iOS, interfejs użytkownika, natywne, kontrolki.	Zrozumienie podstaw tworzenia natywnych interfejsów w Xamarin. Umiejętność stworzenia prostego natywnego interfejsu dla jednej z platform.	Wykorzystanie narzędzi Xamarin do tworzenia natywnych interfejsów dla co najmniej dwóch różnych platform.	Zaawansowane techniki tworzenia natywnych interfejsów, takie jak dostosowywanie kontrolerek do różnych rozdzielczości ekranu i orientacji.	Optymalizacja interfejsu użytkownika w aplikacjach Xamarin dla różnych platform, uwzględniając specyfikę każdego systemu.	Biegłość w tworzeniu skomplikowanych, spersonalizowanych natywnych interfejsów użytkownika dla wszystkich głównych platform mobilnych, jednocześnie dbając o spójność i doskonałe doświadczenie użytkownika.

UWAGI:

1. Ocenę wyższą otrzymuje uczeń spełniający łącznie wymagania edukacyjne określone dla ocen niższych np. ocenę dobrą otrzymuje uczeń spełniający wymagania edukacyjne na oceną dopuszczającą, dostateczną oraz dobrą. W ramach przedmiotu: "Programowania strukturalnego i obiektowego" uczeń powinien wykazać się umiejętnością wykorzystania wiedzy teoretycznej w praktyce.
2. Ocenę niedostateczną otrzymuje uczeń, który nie spełnia wymagań na poszczególne pozytywne oceny.

3. W przypadku nie zrealizowania tematów lekcji (zagadnień) w I okresie będą one realizowane po klasyfikacji śródrocznej. W tym przypadku obowiązują również wymagania edukacyjne dla tych tematów (zagadnień).