

Technikum w Zespole Szkół
im. Armii Krajowej
Obwodu "Głuszec" - Grójec
w Grójcu

Wymagania edukacyjne
na poszczególne oceny szkolne z przedmiotu:
Programowanie Aplikacji Desktopowych

I. Podstawa prawna

1. Ustawa z dnia 7 września 1991 r. o systemie oświaty (tekst jednolity: Dz.U. z 2024 r., poz. 750) - Rozdział 3a
2. Ustawa z dnia 14 grudnia 2016 r. Prawo oświatowe (Dz.U.2023 poz.900)
3. Rozporządzenie Ministra Edukacji Narodowej z dnia 22 lutego 2019 r. w sprawie oceniania, klasyfikowania i promowania uczniów i słuchaczy w szkołach publicznych (tekst jedn.: Dz.U. z 2023 r., poz. 2572)
4. Statut Technikum w Zespole Szkół im. Armii Krajowej Obwodu "Głuszczyce" - Grójec w Grójcu.
5. Program nauczania dla zawodu Technik Programista 351406

II. Wymagania edukacyjne na poszczególne oceny szkolne:

INF.04.5. Programowanie aplikacji desktopowych	
Efekty kształcenia	Kryteria weryfikacji
Uczeń	Uczeń
1) Wykorzystuje środowisko programistyczne dla aplikacji desktopowych	1) dobiera środowisko programistyczne do określonych zadań i języka programowania (np. Visual Studio) 2) stosuje środowiska IDE, RAD 3) rozpoznaje narzędzia wykorzystywane w procesie tworzenia aplikacji desktopowych
2) Wykorzystuje frameworki do programowania aplikacji desktopowych	1) charakteryzuje pojęcie frameworka 2) stosuje frameworki typowe dla aplikacji desktopowych, np. WPF, Qt
3) Programuje desktopowe aplikacje okienkowe	1) rozróżnia elementy interfejsu użytkownika (okno, dialog modalny i niemodalny, kontrolki) 2) stosuje języki programowania odpowiednie dla aplikacji desktopowych, np. C++, C#, Visual Basic, Java, Python 3) stosuje języki do projektowania interfejsu użytkownika np. XAML 4) programuje okna aplikacji

- | | |
|--|---|
| | 5) programuje system menu aplikacji
6) programuje okna dialogowe aplikacji
7) programuje obsługę zdarzeń myszy i klawiatury |
|--|---|

Ocena niedostateczna (1)

- Uczeń nie potrafi poprawnie wymienić podstawowych środowisk programistycznych i narzędzi wykorzystywanych do tworzenia aplikacji desktopowych.
- Uczeń nie rozumie pojęć związanych z frameworkami ani nie potrafi rozróżnić podstawowych elementów interfejsu użytkownika.
- Uczeń nie potrafi skonfigurować środowiska programistycznego ani stworzyć prostego okna aplikacji.
- Uczeń nie jest w stanie zidentyfikować i opisać elementów składających się na prostą aplikację desktopową.
- Uczeń nie potrafi stworzyć działającej aplikacji, łącząc różne elementy interfejsu.
- Uczeń nie potrafi ocenić poprawności działania aplikacji ani wskazać błędów w jej kodzie.

Ocena dopuszczająca (2)

- Uczeń zna podstawowe środowiska programistyczne (np. Visual Studio) i narzędzia używane do tworzenia aplikacji desktopowych.
- Uczeń rozumie podstawowe pojęcia związane z frameworkami i elementami interfejsu użytkownika, ale nie zawsze potrafi je poprawnie zastosować.
- Uczeń potrafi skonfigurować środowisko programistyczne i stworzyć bardzo prostą aplikację okienkową.
- Uczeń potrafi zidentyfikować podstawowe elementy interfejsu (okno, dialog, kontrolki) i wyjaśnić ich funkcje.
- Uczeń potrafi stworzyć prostą aplikację z podstawowymi elementami interfejsu.
- Uczeń potrafi wskazać błędy w działaniu aplikacji, choć ma trudności z ich poprawą.

Ocena dostateczna (3)

- Uczeń zna różne środowiska IDE i RAD oraz narzędzia używane w procesie tworzenia aplikacji desktopowych.
- Uczeń rozumie funkcjonowanie frameworków i potrafi je poprawnie wykorzystać w prostych aplikacjach desktopowych.
- Uczeń potrafi napisać prostą aplikację desktopową z podstawowymi funkcjonalnościami, wykorzystując języki programowania takie jak C++, C#, Java.
- Uczeń potrafi analizować i rozróżniać elementy interfejsu użytkownika oraz stosować podstawowe techniki programowania okienek i dialogów.
- Uczeń potrafi stworzyć aplikację z podstawowym systemem menu oraz oknami dialogowymi, integrując różne komponenty.
- Uczeń potrafi ocenić aplikację pod kątem poprawności działania i wskazać obszary wymagające poprawy.

Ocena dobra (4)

- Uczeń posiada solidną wiedzę na temat różnych środowisk programistycznych i frameworków, w tym WPF i Qt.
- Uczeń dobrze rozumie zasady działania frameworków i języków programowania odpowiednich dla aplikacji desktopowych oraz potrafi to wiedzę zastosować.
- Uczeń potrafi tworzyć bardziej złożone aplikacje desktopowe z zaawansowanym interfejsem użytkownika, wykorzystując języki programowania oraz języki do projektowania interfejsu, takie jak XAML.
- Uczeń potrafi analizować strukturę aplikacji, identyfikując zależności między różnymi elementami interfejsu i kodu.
- Uczeń potrafi stworzyć kompletną aplikację z rozbudowanym systemem menu, oknami dialogowymi i obsługą zdarzeń myszy i klawiatury.
- Uczeń potrafi ocenić działanie aplikacji, wykrywać błędy oraz wprowadzać poprawki i optymalizacje.

Ocena bardzo dobra (5)

- Uczeń posiada bardzo dobrą wiedzę na temat narzędzi, środowisk i frameworków wykorzystywanych w tworzeniu aplikacji desktopowych, potrafi je swobodnie dobierać do różnych zadań.
- Uczeń bardzo dobrze rozumie zaawansowane techniki programowania aplikacji desktopowych, w tym wzorce projektowe i zaawansowane mechanizmy działania frameworków.
- Uczeń potrafi tworzyć złożone i funkcjonalne aplikacje desktopowe, implementując zaawansowane funkcje oraz stosując najlepsze praktyki programistyczne.
- Uczeń potrafi przeprowadzać dogłębną analizę aplikacji, identyfikując potencjalne problemy oraz ich przyczyny.
- Uczeń potrafi projektować i tworzyć zaawansowane aplikacje z rozbudowanym interfejsem, efektywną obsługą zdarzeń i integracją różnych komponentów.
- Uczeń potrafi krytycznie ocenić aplikację, proponować ulepszenia oraz wdrażać zmiany podnoszące jej jakość i funkcjonalność.

Ocena celująca (6)

- Uczeń posiada wszechstronną i głęboką wiedzę na temat wszystkich aspektów tworzenia aplikacji desktopowych, w tym najnowszych narzędzi, technologii i trendów w branży.
- Uczeń doskonale rozumie zarówno teoretyczne, jak i praktyczne aspekty programowania aplikacji desktopowych, w tym zaawansowane wzorce projektowe i architektoniczne.
- Uczeń potrafi tworzyć profesjonalne, złożone aplikacje desktopowe, stosując innowacyjne rozwiązania i nowoczesne technologie.
- Uczeń potrafi przeprowadzać kompleksową analizę aplikacji na każdym etapie jej rozwoju, identyfikując nawet subtelne problemy i proponując nowatorskie rozwiązania.
- Uczeń potrafi tworzyć aplikacje na poziomie eksperckim, integrując zaawansowane funkcjonalności, projektując intuicyjne i wydajne interfejsy użytkownika oraz dbając o jakość kodu.
- Uczeń potrafi ocenić aplikację z punktu widzenia zarówno użytkownika, jak i programisty, wprowadzając zmiany, które znacząco podnoszą jej wartość użytkową oraz technologiczną.