

Wymagania edukacyjne

Pracownia aplikacji desktopowych

Klasa IV i klasa V – technik programista

Sposoby sprawdzania i oceniania osiągnięć edukacyjnych uczniów. Ocenie podlega zarówno wiedza teoretyczna, jak i nabyte w trakcie nauki umiejętności.

Oceniane są:

- Ćwiczenia, zadania wykonywane na lekcji
Ocenie podlega: wykonanie wszystkich poleceń zgodnie z treścią, stopień samodzielności wykonywania zadania, pilność, końcowy efekt pracy (jakość pracy), umiejętność pracy w zespole.
- Aktywność podczas pracy na lekcji.
Ocenie podlega: aktywność ucznia w czasie zajęć, stopień zaangażowania podczas wykonywania zajęć, zainteresowanie tematem lekcji.
- Kartkówki, sprawdziany pisemne lub praktyczne.
Ocena z prac pisemnych zgodna ze Statutem Szkoły
- Zadania dodatkowe, prace projektowe

Wymagania na poszczególne oceny				
dopuszczający	dostateczny	dobry	bardzo dobry	celujący
Środowisko programistyczne				
Potrafi dobrać, zainstalować i skonfigurować środowisko programistyczne do tworzenia aplikacji desktopowych	Potrafi samodzielnie zainstalować i skonfigurować dodatkowe narzędzia i frameworki niezbędne do rozwiązania zadania programistycznego	Potrafi zastosować wybrane biblioteki z funkcjami	Potrafi utworzyć program będący kompletnym rozwiązaniem	Potrafi przygotować swoje biblioteki klas Tworzy programy będące kreatywnym rozwiązaniem zaawansowanych problemów

Tworzenie aplikacji desktopowych				
<p>Potrafi wykorzystać język XAML do opisu interfejsu użytkownika (UI)</p> <p>Potrafi utworzyć podstawowe kontrolki interfejsu użytkownika</p> <p>Potrafi ustawić podstawowe własności kontrolek UI</p> <p>Potrafi napisać obsługę podstawowych zdarzeń</p> <p>Potrafi wykorzystać podstawowe układy elementów w aplikacji</p>	<p>Potrafi odpowiednio wykorzystać kontrolki do utworzenia UI</p> <p>Potrafi właściwie ustawić najczęściej używane własności elementów UI</p> <p>Odpowiednio dobiera zdarzenia w zależności od potrzeb</p> <p>Umie napisać obsługę zdarzeń podstawowych elementów UI</p> <p>Potrafi dobrać i zastosować właściwy układ elementów w interfejsie użytkownika</p>	<p>Umie odpowiednio dobrać własności elementów UI</p> <p>Umie odpowiednio dobrać i skonfigurować elementy interfejsu użytkownika</p> <p>Umie obsłużyć zdarzenia wraz z walidacją danych wejściowych</p> <p>Potrafi odpowiednio dobrać układy elementów w UI</p> <p>Potrafi skorzystać z dokumentacji języka programowania</p>	<p>Potrafi przygotować interfejs użytkownika zgodnie z wymaganiami projektu</p> <p>Potrafi napisać obsługę wszystkich zdarzeń generowanych przez elementy interfejsu użytkownika</p> <p>Potrafi samodzielnie rozwiązać zadanie</p> <p>Potrafi znaleźć właściwe informacje w dokumentacji języka programowania</p>	<p>Potrafi samodzielnie wykorzystać elementy interfejsu użytkownika do tworzenia aplikacji</p> <p>Potrafi samodzielnie wykorzystywać dokumentację do rozwijania swojej wiedzy</p> <p>Biegłe posługuje się wiadomościami w rozwiązywaniu problemów teoretycznych i praktycznych</p>
Język programowania				
<p>Potrafi zastosować proste typy danych w swoich programach</p> <p>Potrafi utworzyć proste klasy i metody</p>	<p>W swoich programach dobiera typy danych odpowiednio do problemu</p> <p>Wykorzystuje operacje na prostych typach danych</p> <p>Potrafi utworzyć klasy zgodnie z wymaganiami</p>	<p>Umie zadeklarować swoje typy danych</p> <p>Potrafi zastosować obiektowe podejście do rozwiązania zagadnienia</p>	<p>Potrafi wykorzystać mechanizm dziedziczenia i utworzyć klasy bazowe oraz pochodne</p> <p>Potrafi utworzyć szablony klasy</p>	<p>Potrafi obsłużyć wyjątki w swoim</p> <p>Potrafi kreatywnie rozwiązać problem z zastosowaniem optymalnych rozwiązań</p> <p>Biegłe posługuje się wiadomościami w</p>

Potrafi napisać prostą metodę obsługi zdarzenia	Wykorzystuje gotowe klasy szablonowe	Potrafi zastosować proste mechanizmy obiektowe w swojej aplikacji Wykorzystuje gotowe klasy szablonowe i interfejsy		rozwiązywaniu problemów teoretycznych i praktycznych
---	--------------------------------------	--	--	--