

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

Wymagania edukacyjne
na poszczególne oceny szkolne z przedmiotu:
Testowanie i Dokumentowanie Aplikacji

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.04.3. Testowanie i dokumentowanie aplikacji	
Efekty kształcenia	Kryteria weryfikacji
Uczeń	Uczeń
1) przeprowadza walidację kodu programu	1) dobiera narzędzia i środowisko do testowania programów; 2) wykonuje testy tworzonych programów; 3) identyfikuje błędy w tworzonych programach; 4) poprawia błędy w tworzonych programach, skryptach; 5) optymalizuje kod źródłowy
2) dokumentuje tworzoną aplikację	1) stosuje komentarze w kodzie źródłowym programu; 2) pisze dokumentację kodu; 3) pisze dokumenty pomocy programu; 4) pisze instrukcję użytkownika aplikacji; 5) pisze instrukcję użytkownika programu; 6) pisze dokumentację wdrożenia projektu; 7) przygotowuje dokumentację z wykonanych testów
3) przeprowadza testy	1) charakteryzuje metodyki testowania; 2) przygotowuje testy funkcjonalne i нефункционалне;

- | | |
|--|---|
| | <ol style="list-style-type: none">3) przeprowadza testy funkcjonalne;4) przeprowadza testy w kolejnych fazach projektu informatycznego;5) przeprowadza testy нефункционалне: użyteczności, wydajnościowe, obciążeniowe, zgodności, bezpieczeństwa;6) korzysta z systemów raportowania błędów, np. BugZilla, JIRA;7) przeprowadza testy interfejsu;8) testuje prototyp projektu interfejsu;9) przygotowuje środowiska testowe;10) planuje scenariusze testowania aplikacji;11) pisze scenariusze testowania aplikacji;12) raportuje błędy po przeprowadzonych testach;13) automatyzuje proces testowania;14) testuje aplikację;15) weryfikuje zgodność aplikacji z oczekiwaniami klienta |
|--|---|

Ocena niedostateczna (1)

- Uczeń nie zna podstawowych narzędzi i metod testowania aplikacji oraz nie potrafi opisać, czym jest walidacja kodu.
- Uczeń nie rozumie znaczenia testowania i dokumentowania aplikacji oraz nie potrafi wytłumaczyć metod testowania funkcjonalnego czy нефункционалнеgo.
- Uczeń nie potrafi przeprowadzić testów aplikacji ani napisać komentarzy w kodzie źródłowym.
- Uczeń nie potrafi zidentyfikować błędów w programie i nie jest w stanie przeprowadzić walidacji kodu ani testów.
- Uczeń nie podejmuje prób tworzenia dokumentacji kodu ani nie przygotowuje scenariuszy testowych.
- Uczeń nie potrafi ocenić jakości aplikacji ani przygotować dokumentacji z testów.

Ocena dopuszczająca (2)

- Uczeń zna podstawowe narzędzia do testowania aplikacji, ale ma trudności z ich stosowaniem.
- Uczeń rozumie potrzebę testowania i dokumentowania aplikacji, ale ma problemy z właściwym opisem metod testowania (funkcjonalnego, нефункционального).
- Uczeń potrafi przeprowadzić podstawowe testy aplikacji, dodać proste komentarze w kodzie i sporządzić ogólną dokumentację.
- Uczeń potrafi zidentyfikować niektóre błędy w programach i przeprowadzić podstawową walidację kodu.
- Uczeń podejmuje próby przygotowania prostych scenariuszy testowania, ale wymaga wsparcia w ich realizacji i dokumentowaniu.
- Uczeń potrafi wskazać błędy po przeprowadzonych testach, ale nie proponuje skutecznych rozwiązań.

Ocena dostateczna (3)

- Uczeń zna narzędzia i środowiska do testowania programów oraz podstawowe techniki testowania aplikacji, takie jak testy funkcjonalne i нефункциональные.
- Uczeń rozumie proces testowania i potrafi opisać różne metodyki testowania oraz znaczenie walidacji kodu.
- Uczeń potrafi przeprowadzić testy aplikacji, sporządzić dokumentację kodu, a także napisać prostą instrukcję użytkownika i scenariusz testowy.
- Uczeń potrafi analizować wyniki testów, identyfikować błędy w programie oraz optymalizować kod źródłowy.
- Uczeń potrafi przygotować dokumentację z przeprowadzonych testów oraz napisać prostą instrukcję użytkownika aplikacji.
- Uczeń potrafi ocenić wyniki testów, raportować błędy i zaproponować poprawki w kodzie.

Ocena dobra (4)

- Uczeń posiada solidną wiedzę na temat metod testowania aplikacji, rozumie narzędzia raportowania błędów (np. JIRA, BugZilla) oraz zasady automatyzacji testów.
- Uczeń dobrze rozumie proces testowania aplikacji, potrafi charakteryzować różne rodzaje testów (np. wydajnościowe, zgodności, użyteczności) oraz zna zasady pisania dokumentacji wdrożenia projektu.
- Uczeń potrafi przeprowadzić zaawansowane testy aplikacji, optymalizować kod, tworzyć kompletne scenariusze testowe oraz dokumentację użytkownika.
- Uczeń potrafi dokładnie analizować wyniki testów, identyfikować problemy w aplikacji oraz sugerować optymalne rozwiązania.
- Uczeń potrafi tworzyć szczegółową dokumentację z przeprowadzonych testów, przygotować dokumentację wdrożenia projektu oraz instrukcje użytkownika aplikacji.

- Uczeń potrafi krytycznie ocenić zgodność aplikacji z wymaganiami klienta i efektywność przeprowadzonych testów.

Ocena bardzo dobra (5)

- Uczeń posiada bardzo dobrą znajomość narzędzi testowych oraz rozumie metodyki automatyzacji testów i stosowania systemów raportowania błędów.
- Uczeń bardzo dobrze rozumie cały proces testowania, w tym metodyki testowania aplikacji w kolejnych fazach projektu oraz znaczenie testowania interfejsu i prototypu projektu.
- Uczeń potrafi przeprowadzić zaawansowane testy нефункционалне (np. obciążeniowe, bezpieczeństwa) oraz stworzyć kompletne, profesjonalne instrukcje użytkownika, dokumentację techniczną i wdrożeniową.
- Uczeń potrafi przeprowadzić zaawansowaną analizę wyników testów, identyfikować błędy i ich przyczyny oraz optymalizować aplikacje pod kątem wydajności.
- Uczeń potrafi projektować i automatyzować proces testowania, raportować błędy oraz tworzyć zaawansowane scenariusze testowe.
- Uczeń potrafi ocenić jakość aplikacji na każdym etapie projektu, proponując zmiany mające na celu poprawę wydajności i zgodności aplikacji z oczekiwaniami klienta.

Ocena celująca (6)

- Uczeń posiada wszechstronną wiedzę na temat metod testowania, dokumentowania oraz automatyzacji testów, zna najnowsze technologie i narzędzia stosowane w testowaniu aplikacji.
- Uczeń doskonale rozumie każdy aspekt procesu testowania, w tym zaawansowane techniki optymalizacji kodu oraz automatyzacji testowania.
- Uczeń potrafi samodzielnie przeprowadzać kompleksowe testy aplikacji, w tym testy bezpieczeństwa, zgodności oraz optymalizować aplikację na najwyższym poziomie.
- Uczeń potrafi przeprowadzać dogłębną analizę wyników testów, przewidywać problemy i rozwiązywać złożone błędy oraz optymalizować proces testowania.
- Uczeń potrafi tworzyć kompleksowe strategie testowe i dokumentacje, obejmujące wszystkie aspekty aplikacji, od testów funkcjonalnych po wdrożenie i optymalizację.
- Uczeń potrafi krytycznie ocenić całość projektu, zgodność z oczekiwaniami klienta, jakość aplikacji i efektywność procesu testowania, wdrażając innowacyjne rozwiązania.