

Wymagania edukacyjne

Podstawy programowania (Python + Tkinter)

Zawód: technik programista (technikum 5-letnie)

Kwalifikacja: INF.04. Projektowanie, programowanie i testowanie aplikacji

Przedmioty: Podstawy programowania

Klasa: II

Źródła: *Program nauczania wydawnictwa Helion — technik programista* (dział 3.2. Podstawy programowania, dział 3.8. Pracownia podstaw programowania), *Podstawa programowa kształcenia w zawodach* (Dziennik Ustaw, poz. 991) — efekty kształcenia i kryteria weryfikacji ujęte w programie nauczania jako część jednostki INF.04.3. Projektowanie oprogramowania, dotycząca środowisk programistycznych, frameworków i programowania aplikacji desktopowych.

Zasady konstrukcji wymagań

1. Przedmiot **Pracownia podstaw programowania** (praktyka) jest oceniany na podstawie **działającej aplikacji** — uczeń tworzy, implementuje, uruchamia, testuje i poprawia programy okienkowe w środowisku IDE, zgodnie z zaleceniami programu nauczania dla zajęć praktycznych.

2. **Zasada kumulatywności ocen.**

Uzyskanie oceny wyższej wymaga opanowania wiadomości i umiejętności przypisanych do wszystkich ocen niższych. Np. na ocenę **dobrą (4)** uczeń musi dodatkowo spełniać wymagania na oceny **dopuszczającą (2)** i **dostateczną (3)**. Ocena **celująca (6)** oznacza pełne (100%), bezbłędne i w pełni samodzielne opanowanie wymagań przypisanych do ocen 2–5 w danym dziale.

3. **Adaptacja do języka Python i biblioteki Tkinter.**

Program nauczania i podstawa programowa formułują efekty kształcenia niezależnie od konkretnego języka i frameworka (przykłady: C++, C#, Visual Basic, Java, Python; WPF, Qt; XAML). Poniższe wymagania dostosowano do Pythona z biblioteką **Tkinter**:

- *środowisko programistyczne* (np. *Visual Studio*, *CodeBlocks*) → środowiska dla Pythona, np. **PyCharm**, **Visual Studio Code**;
- *framework aplikacji desktopowych* (np. *WPF*, *Qt*) → **Tkinter** — biblioteka wchodząca w skład standardowej biblioteki Pythona, oparta na Tk/Tcl; dla porównania teoretycznego wspomniane są także inne biblioteki GUI dla Pythona (np. *PyQt*, *wxPython*);
- *język do projektowania interfejsu użytkownika* (np. *XAML*) → w Tkinter interfejs definiuje się bezpośrednio w kodzie Pythona, za pomocą menedżerów geometrii **pack**, **grid**, **place**, bez oddzielnego języka znaczników;

- *zdarzenia myszy i klawiatury* → mechanizm wiązania zdarzeń w Tkinter (bind, parametr command widżetu) z funkcją obsługi zdarzenia.
4. Godziny zajęć wg planu nauczania: **Pracownia podstaw programowania** (praktyka) — klasa II: 2 godz./tydz. (60 godz.).
 5. **Ocena niedostateczna.**

We wszystkich działach ocenę niedostateczną otrzymuje uczeń, który nie spełnia wymagań na ocenę dopuszczającą — nie posiada wiedzy/umiejętności koniecznych do kontynuowania nauki nawet przy pomocy nauczyciela, nie podejmuje samodzielnej pracy lub oddaje prace niesamodzielne.

KLASA II

Pracownia podstaw programowania

Dział 1. Konfiguracja środowiska i uruchamianie aplikacji okienkowych w Tkinter

Ocena	Uczeń wykonuje
Dopuszczająca (2)	Tworzy i uruchamia gotowy, prosty program Tkinter według instrukcji (jedno okno z jednym widżetem); zapisuje plik źródłowy .py.
Dostateczna (3)	Samodzielnie tworzy prostą aplikację okienkową wyświetlającą okno z etykietą i przyciskiem; uruchamia aplikację i poprawia podstawowe błędy składniowe.
Dobra (4)	Konfiguruje wybrane środowisko IDE do pracy z Pythonem i Tkinter; korzysta z komunikatów błędów lub debugera do lokalizowania i poprawiania błędów w kodzie aplikacji okienkowej.
Bardzo dobra (5)	Testuje działanie aplikacji okienkowej dla różnych sposobów interakcji użytkownika (kliknięcia, wprowadzane dane); samodzielnie diagnozuje i usuwa przyczyny nieprawidłowego działania interfejsu.
Celująca (6)	Bezbłędnie i w pełni samodzielnie konfiguruje środowisko oraz tworzy, uruchamia i debuguje aplikacje okienkowe przewidziane programem działu, organizując kod w czytelny sposób (np. w osobnych funkcjach obsługi zdarzeń).

Dział 2. Implementacja interfejsu z wykorzystaniem widżetów i menedżerów geometrii

Ocena	Uczeń wykonuje
-------	----------------

Dopuszczająca (2)	Umieszcza na oknie pojedynczy widżet (np. Label, Button) według wzoru, korzystając ze wskazanego menedżera geometrii.
Dostateczna (3)	Tworzy okno z kilkoma widżetami (Label, Button, Entry) rozmieszczonymi za pomocą pack lub grid; odczytuje i wyświetla dane wprowadzone przez użytkownika w polu Entry.
Dobra (4)	Samodzielnie dobiera i stosuje menedżer geometrii (pack, grid lub place) odpowiedni do układu okna; tworzy interfejs złożony z kilku współpracujących ze sobą widżetów (np. prosty formularz).
Bardzo dobra (5)	Projektuje i implementuje spójny, czytelny interfejs użytkownika dla opisanego zadania (np. formularz z ocenami uczniów); testuje działanie interfejsu dla różnych danych wejściowych, w tym błędnych.
Celująca (6)	Bez błędnie i w pełni samodzielnie implementuje rozbudowany interfejs łączący kilka menedżerów geometrii i wiele współpracujących widżetów, zgodnie z pełnym zakresem wymagań działu.

Dział 3. Implementacja okien, menu, okien dialogowych i obsługi zdarzeń — projekt aplikacji okienkowej

Przykładowe tematy zadań/projektów (wskazane wprost w programie nauczania [1]): aplikacja do wprowadzania i uśredniania ocen z przedmiotu dla wielu uczniów, wskazująca ucznia z najwyższą i najniższą średnią; aplikacja z menu tworzonym w oparciu o Tkinter; kalkulator wykonujący dodawanie, odejmowanie, mnożenie i dzielenie na dwóch liczbach wprowadzonych przez użytkownika.

Ocena	Uczeń wykonuje
Dopuszczająca (2)	Dodaje do aplikacji pojedynczy element menu lub prosty komunikat (okno dialogowe) według wzoru.
Dostateczna (3)	Implementuje prostą procedurę obsługi zdarzenia (np. kliknięcia przycisku) wykonującą wskazane działanie; tworzy proste menu aplikacji z kilkoma pozycjami.
Dobra (4)	Implementuje aplikację z oknem dialogowym (modalnym lub niemodalnym) powiązanym z oknem głównym; obsługuje zdarzenia myszy i klawiatury dla kilku widżetów jednocześnie.
Bardzo dobra (5)	Implementuje kompletną aplikację okienkową łączącą menu, okno dialogowe oraz obsługę zdarzeń myszy i klawiatury (np. kalkulator wykonujący cztery działania lub aplikację do wprowadzania i uśredniania ocen); testuje działanie aplikacji dla różnych danych, w tym błędnych.

Celująca (6)	Realizuje w 100%, bezbłędnie i w pełni samodzielnie kompletny projekt aplikacji okienkowej (na bazie przykładowych zadań programu nauczania) z menu, oknami dialogowymi i pełną obsługą zdarzeń; testuje aplikację dla różnych przypadków i dba o czytelność oraz spójność interfejsu.
---------------------	--