



# MONITOR POLSKI

DZIENNIK URZĘDOWY RZECZYPOSPOLITEJ POLSKIEJ

---

Warszawa, dnia 27 czerwca 2024 r.

Poz. 544

**OBWIESZCZENIE  
MINISTRA CYFRYZACJI<sup>1)</sup>**

z dnia 7 czerwca 2024 r.

**w sprawie włączenia kwalifikacji wolnorynkowej „Programowanie w języku Python”  
do Zintegrowanego Systemu Kwalifikacji**

Na podstawie art. 25 ust. 1 i 2 ustawy z dnia 22 grudnia 2015 r. o Zintegrowanym Systemie Kwalifikacji (Dz. U. z 2020 r. poz. 226 oraz z 2023 r. poz. 2005) ogłasza się w załączniku do niniejszego obwieszczenia informacje o włączeniu kwalifikacji wolnorynkowej „Programowanie w języku Python” do Zintegrowanego Systemu Kwalifikacji.

Minister Cyfryzacji: *K. Gawkowski*

---

<sup>1)</sup> Minister Cyfryzacji kieruje działem administracji rządowej – informatyzacja, na podstawie § 1 ust. 2 rozporządzenia Prezesa Rady Ministrów z dnia 18 grudnia 2023 r. w sprawie szczegółowego zakresu działania Ministra Cyfryzacji (Dz. U. poz. 2720).

Załącznik do obwieszczenia Ministra Cyfryzacji  
z dnia 7 czerwca 2024 r. (M.P. poz. 544)

INFORMACJE O WŁĄCZENIU KWALIFIKACJI WOLNORYNKOWEJ „PROGRAMOWANIE W JĘZYKU PYTHON”  
DO ZINTEGROWANEGO SYSTEMU KWALIFIKACJI

**1. Nazwa kwalifikacji wolnorynkowej**

Programowanie w języku Python

**2. Poziom Polskiej Ramy Kwalifikacji przypisany do kwalifikacji wolnorynkowej**

4 poziom Polskiej Ramy Kwalifikacji

**3. Efekty uczenia się wymagane dla kwalifikacji wolnorynkowej**

**Syntetyczna charakterystyka efektów uczenia się**

Osoba posiadająca kwalifikację wolnorynkową jest gotowa do samodzielnego tworzenia oprogramowania w języku Python. W swojej pracy wykorzystuje zarówno podstawowe elementy języka Python, jak i programowania obiektowego. Realizując zadania zawodowe, posługuje się specjalistyczną wiedzą z zakresu wytwarzania oprogramowania, a ponadto podejmuje działania pozwalające na przetestowanie oraz usunięcie błędów pojawiających się w programie. Jest gotowa ponosić odpowiedzialność za wykonywanie samodzielnych zadań zawodowych.

**Zestaw 1. Programowanie z wykorzystaniem podstawowych elementów języka Python**

Poszczególne efekty uczenia się	Kryteria weryfikacji ich osiągnięcia
Wykorzystuje środowiska programistyczne do tworzenia programu	<ul style="list-style-type: none"> <li>– charakteryzuje zadania interpretera i debuggera,</li> <li>– analizuje błędy w kodzie za pomocą debuggera,</li> <li>– charakteryzuje pojęcie biblioteki,</li> <li>– uruchamia stworzone programy.</li> </ul>
Posługuje się prostymi typami danych	<ul style="list-style-type: none"> <li>– rozróżnia typy liczbowe całkowite i zmiennoprzecinkowe,</li> <li>– rozpoznaje typ logiczny,</li> <li>– charakteryzuje pojęcie typu string,</li> <li>– wykorzystuje typy liczbowe całkowite, zmiennoprzecinkowe, znakowe, typ logiczny i typ łańcuchowy.</li> </ul>
Korzysta z operacji wejścia i wyjścia	<ul style="list-style-type: none"> <li>– korzysta z różnych funkcji do operacji wejścia i wyjścia,</li> <li>– posługuje się funkcjami print oraz input.</li> </ul>
Posługuje się instrukcjami sterującymi	<ul style="list-style-type: none"> <li>– rozróżnia instrukcje warunkowe: if oraz if else,</li> <li>– rozróżnia pętle: for oraz while,</li> <li>– korzysta z instrukcji: if, if else, for, while, break, continue.</li> </ul>
Posługuje się złożonymi typami danych	<ul style="list-style-type: none"> <li>– rozróżnia rodzaje złożonych typów danych,</li> <li>– korzysta z list oraz tuples.</li> </ul>
Posługuje się operatorami	<ul style="list-style-type: none"> <li>– rozróżnia operatory logiczne: and, or, not,</li> <li>– rozpoznaje operatory przesunięcia bitowego,</li> <li>– wykorzystuje operatory logiczne, operatory przesunięcia bitowego oraz operatory bitowe.</li> </ul>
Stosuje własne funkcje	<ul style="list-style-type: none"> <li>– rozpoznaje przekazywanie parametrów do funkcji,</li> <li>– tworzy własne funkcje,</li> <li>– posługuje się rekurencyjnym wywołaniem funkcji.</li> </ul>

Wykonuje operacje odczytywania i zapisywania plików	<ul style="list-style-type: none"> <li>– rozróżnia pliki tekstowe oraz binarne,</li> <li>– posługuje się typem plikowym,</li> <li>– odczytuje dane z pliku,</li> <li>– zapisuje dane do pliku.</li> </ul>
---	---

<b>Zestaw 2. Programowanie obiektowe</b>	
<b>Poszczególne efekty uczenia się</b>	<b>Kryteria weryfikacji ich osiągnięcia</b>
Stosuje zasady programowania obiektowego	<ul style="list-style-type: none"> <li>– charakteryzuje pojęcia: klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm,</li> <li>– dzieli zagadnienie na klasy,</li> <li>– powołuje obiekty,</li> <li>– projektuje aplikację z zastosowaniem zasad programowania obiektowego.</li> </ul>
Korzysta z klas i obiektów	<ul style="list-style-type: none"> <li>– definiuje pola klasy,</li> <li>– definiuje metody klasy,</li> <li>– definiuje konstruktory,</li> <li>– tworzy obiekty,</li> <li>– posługuje się składowymi obiektów.</li> </ul>
Definiuje klasy bazowe i klasy pochodne	<ul style="list-style-type: none"> <li>– buduje hierarchię dziedziczenia klas w programie,</li> <li>– wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia,</li> <li>– definiuje klasy pochodne.</li> </ul>
Programuje obsługę wyjątków	<ul style="list-style-type: none"> <li>– stosuje szkielet obsługi wyjątków z instrukcjami try i except,</li> <li>– stosuje instrukcję raise,</li> <li>– opracowuje listę możliwych błędów wykonania aplikacji,</li> <li>– definiuje obsługę błędów w aplikacji.</li> </ul>

<b>Zestaw 3. Korzystanie z kolekcji danych oraz wątków</b>	
<b>Poszczególne efekty uczenia się</b>	<b>Kryteria weryfikacji ich osiągnięcia</b>
Korzysta z kolekcji danych oraz wątków	<ul style="list-style-type: none"> <li>– posługuje się kolekcją: dictionary,</li> <li>– stosuje kolekcję: deque,</li> <li>– charakteryzuje cechy kolekcji,</li> <li>– posługuje się wyrażeniami Lambda.</li> </ul>
Korzysta z mechanizmu współbieżności	<ul style="list-style-type: none"> <li>– posługuje się biblioteką thread,</li> <li>– posługuje się biblioteką multiprocessing,</li> <li>– stosuje mechanizm synchronizacji wątków,</li> <li>– charakteryzuje problem zakleszczania wątków.</li> </ul>

#### **4. Ramowe wymagania dotyczące metod przeprowadzania walidacji, osób przeprowadzających walidację oraz warunków organizacyjnych i materialnych niezbędnych do prawidłowego i bezpiecznego przeprowadzania walidacji**

<p><b>1. Etap weryfikacji</b></p> <p><b>1.1. Metody</b></p> <p>W części teoretycznej wykorzystuje się metodę testu teoretycznego.</p> <p>W części praktycznej wykorzystuje się metodę obserwacji w warunkach symulowanych uzupełnioną wywiadem swobodnym (rozmową z komisją).</p> <p>Pozytywny wynik z części teoretycznej jest warunkiem przystąpienia do części praktycznej.</p> <p>Możliwe jest zastosowanie innych metod walidacji lub jej form (np. walidacja w formie zdalnej) przy zachowaniu wszelkich zasad zapewniania jakości oraz obowiązku sprawdzenia wszystkich efektów uczenia się wraz z kryteriami weryfikacji.</p>
---

**1.2. Zasoby kadrowe**

Weryfikację efektów uczenia się prowadzi komisja walidacyjna składająca się z minimum 2 osób spełniających następujące warunki:

asesor – posiada ukończone studia wyższe na kierunku informatyka lub pokrewnym (co najmniej 6 poziom Polskiej Ramy Kwalifikacji), minimum 5 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela programowania

oraz

przewodniczący komisji walidacyjnej – posiada ukończone studia wyższe na kierunku informatyka lub pokrewnym (co najmniej 7 poziom Polskiej Ramy Kwalifikacji), minimum 10 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela programowania.

**1.3. Sposób organizacji walidacji oraz warunki organizacyjne i materialne**

W przypadku walidacji w formie stacjonarnej, instytucja prowadząca walidację zapewnia:

- 1) stanowiska komputerowe (jedno stanowisko dla jednego kandydata) połączone z Internetem, wyposażone w:
  - a) system operacyjny z interfejsem graficznym,
  - b) przeglądarkę internetową,
  - c) edytor tekstu lub środowisko programistyczne zapewniające możliwość tworzenia programów w języku Python (np. PyCharm);
- 2) materiały biurowe (kartki, długopisy).

W przypadku zdalnego prowadzenia walidacji komisja zatwierdza warunki przystąpienia do walidacji, mając na uwadze możliwości techniczne. Użyty podczas walidacji system teleinformatyczny powinien gwarantować komisji walidacyjnej możliwość stałej obserwacji kandydata, w szczególności identyfikację osoby, która przystępuje do walidacji, samodzielność jej pracy i zabezpieczenie przebiegu walidacji przed ingerencją osób trzecich, to jest zapewnić możliwość wiarygodnego sprawdzenia, czy osoba ubiegająca się o nadanie kwalifikacji rynkowej osiągnęła wyodrębnioną część lub całość efektów uczenia się wymaganych dla tej kwalifikacji.

Osoby, które przystąpiły do walidacji, powinny utworzyć program w języku Python według wskazanych założeń obejmujących podstawowe elementy języka, klasy i metody, kolekcje, wątki i obsługę sieci. Możliwe jest korzystanie przez kandydata z materiałów dodatkowych w postaci literatury lub przykładowych fragmentów kodu znalezionych w sieci Internet.

**2. Etap identyfikowania i dokumentowania efektów uczenia się**

Nie określa się wymagań dotyczących etapów identyfikowania i dokumentowania efektów uczenia się.

**5. Warunki, jakie musi spełniać osoba przystępująca do walidacji**

Brak warunków.

**6. Inne, poza pozytywnym wynikiem walidacji, warunki uzyskania kwalifikacji wolnorynkowej**

Brak innych, poza pozytywnym wynikiem walidacji, warunków uzyskania kwalifikacji wolnorynkowej.

**7. Okres ważności certyfikatu kwalifikacji wolnorynkowej**

Bezterminowo.

**8. Termin dokonywania przeglądu kwalifikacji wolnorynkowej**

Nie rzadziej niż raz na 10 lat.