

Warszawa, dnia 27 czerwca 2024 r.

Poz. 544

**OBWIESZCZENIE
MINISTRA CYFRYZACJI¹⁾**

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*

¹⁾ 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"> – charakteryzuje zadania interpretera i debuggera, – analizuje błędy w kodzie za pomocą debuggera, – charakteryzuje pojęcie biblioteki, – uruchamia stworzone programy.
Posługuje się prostymi typami danych	<ul style="list-style-type: none"> – rozróżnia typy liczbowe całkowite i zmiennoprzecinkowe, – rozpoznaje typ logiczny, – charakteryzuje pojęcie typu string, – wykorzystuje typy liczbowe całkowite, zmiennoprzecinkowe, znakowe, typ logiczny i typ łańcuchowy.
Korzysta z operacji wejścia i wyjścia	<ul style="list-style-type: none"> – korzysta z różnych funkcji do operacji wejścia i wyjścia, – posługuje się funkcjami print oraz input.
Posługuje się instrukcjami sterującymi	<ul style="list-style-type: none"> – rozróżnia instrukcje warunkowe: if oraz if else, – rozróżnia pętle: for oraz while, – korzysta z instrukcji: if, if else, for, while, break, continue.
Posługuje się złożonymi typami danych	<ul style="list-style-type: none"> – rozróżnia rodzaje złożonych typów danych, – korzysta z list oraz tuples.
Posługuje się operatorami	<ul style="list-style-type: none"> – rozróżnia operatory logiczne: and, or, not, – rozpoznaje operatory przesunięcia bitowego, – wykorzystuje operatory logiczne, operatory przesunięcia bitowego oraz operatory bitowe.
Stosuje własne funkcje	<ul style="list-style-type: none"> – rozpoznaje przekazywanie parametrów do funkcji, – tworzy własne funkcje, – posługuje się rekurencyjnym wywołaniem funkcji.

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

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

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

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

<p>1. Etap weryfikacji</p> <p>1.1. Metody</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.