

Warszawa, dnia 20 czerwca 2023 r.

Poz. 587

**OBWIESZCZENIE
MINISTRA CYFRYZACJI¹⁾**

z dnia 18 maja 2023 r.

w sprawie włączenia kwalifikacji rynkowej „Programowanie w języku Java” 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) ogłasza się w załączniku do niniejszego obwieszczenia informacje o włączeniu kwalifikacji rynkowej „Programowanie w języku Java” do Zintegrowanego Systemu Kwalifikacji.

Minister Cyfryzacji: *J. Cieszyński*

¹⁾ Minister Cyfryzacji kieruje działem administracji rządowej – informatyzacja, na podstawie § 1 ust. 2 rozporządzenia Prezesa Rady Ministrów z dnia 26 kwietnia 2023 r. w sprawie szczegółowego zakresu działania Ministra Cyfryzacji (Dz. U. poz. 792).

Załącznik do obwieszczenia Ministra Cyfryzacji
z dnia 18 maja 2023 r. (M.P. poz. 587)

INFORMACJE O WŁĄCZENIU KWALIFIKACJI RYNKOWEJ „PROGRAMOWANIE W JĘZYKU JAVA”
DO ZINTEGROWANEGO SYSTEMU KWALIFIKACJI

1. Nazwa kwalifikacji rynkowej

Programowanie w języku Java

2. Nazwa dokumentu potwierdzającego nadanie kwalifikacji rynkowej

Certyfikat

3. Okres ważności dokumentu potwierdzającego nadanie kwalifikacji rynkowej

Bezterminowo

4. Poziom Polskiej Ramy Kwalifikacji przypisany do kwalifikacji rynkowej

4 poziom Polskiej Ramy Kwalifikacji

5. Efekty uczenia się wymagane dla kwalifikacji rynkowej

Syntetyczna charakterystyka efektów uczenia się

Osoba posiadająca kwalifikację jest gotowa do samodzielnego tworzenia oprogramowania w języku Java. W swojej pracy wykorzystuje zarówno podstawowe elementy języka Java jak i programowania obiektowego. Korzysta z kolekcji, wątków, obsługi sieci i baz danych. 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 przygotowana do pracy w zmiennych warunkach.

Zestaw 1. Podstawowe elementy języka Java

Poszczególne efekty uczenia się	Kryteria weryfikacji ich osiągnięcia
1. Wykorzystuje środowiska programistyczne do tworzenia programów	a) charakteryzuje zadania kompilatora i debuggera, b) analizuje błędy w kodzie za pomocą debuggera, c) charakteryzuje pojęcie biblioteki, d) kompiluje i uruchamia stworzone programy.
2. Posługuje się prostymi typami danych	a) rozróżnia typy liczbowe całkowite i zmiennoprzecinkowe, b) rozpoznaje typ logiczny, c) rozróżnia typy znakowe i łańcuchowe, d) wykorzystuje typy liczbowe całkowite, zmiennoprzecinkowe, znakowe, typ logiczny i typ łańcuchowy.
3. Korzysta z operacji wejścia i wyjścia	a) korzysta z różnych funkcji do operacji wejścia i wyjścia, b) posługuje się obiektem klasy Scanner.
4. Posługuje się instrukcjami sterującymi	a) rozróżnia instrukcje warunkowe if oraz switch, b) rozróżnia pętle: for, while oraz do-while, c) korzysta z instrukcji if, for, break, continue.
5. Posługuje się złożonymi typami danych	a) rozróżnia rodzaje złożonych typów danych, b) korzysta z tablic jednowymiarowych i wielowymiarowych, c) rozróżnia referencję od zmiennej.
6. Posługuje się operatorami	a) rozróżnia operatory logiczne, w tym && – and, – or, ! – not, b) rozpoznaje operatory przesunięcia bitowego, c) wykorzystuje operatory logiczne, operatory przesunięcia bitowego oraz operatory bitowe, w tym AND, OR, XOR, NOT.

7. Stosuje własne metody	<ul style="list-style-type: none"> a) rozróżnia różne sposoby przekazywania argumentów do metod, b) rozpoznaje przekazywanie parametrów do metod przez wartość oraz referencję, c) tworzy własne metody, d) posługuje się rekurencyjnym wywołaniem metod.
8. Wykonuje operacje odczytywania i zapisywania plików	<ul style="list-style-type: none"> a) rozróżnia pliki tekstowe oraz binarne, b) posługuje się typem plikowym, c) odczytuje i zapisuje dane.

Zestaw 2. Programowanie obiektowe

Poszczególne efekty uczenia się	Kryteria weryfikacji ich osiągnięcia
1. Stosuje zasady programowania obiektowego	<ul style="list-style-type: none"> a) charakteryzuje pojęcia: klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm, b) dzieli zagadnienie na klasy, c) powołuje obiekty, d) projektuje aplikację z zastosowaniem hermetyzacji, dziedziczenia i polimorfizmu.
2. Korzysta z klas i obiektów	<ul style="list-style-type: none"> a) definiuje pola klasy, b) określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu, c) definiuje metody klasy, d) definiuje konstruktory, w tym konstruktor kopiujący, e) określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu, f) deklaruje obiekty i odwołuje się obiektem do składowych klasy, g) definiuje składniki statyczne klasy, h) stosuje składnik statyczny klasy i metody do ich obsługi.
3. Definiuje klasy abstrakcyjne, interfejsy i klasy pochodne	<ul style="list-style-type: none"> a) buduje hierarchię dziedziczenia klas w programie, b) wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia, c) definiuje klasy bazowe i pochodne, d) definiuje klasy abstrakcyjne i interfejsy.
4. Programuje obsługę wyjątków	<ul style="list-style-type: none"> a) stosuje szkielet obsługi wyjątków z instrukcjami try i catch, b) stosuje instrukcję throw, c) opracowuje listę możliwych błędów wykonania aplikacji, d) definiuje obsługę dla błędów wykonania aplikacji w wyniku wykonywania różnych operacji.

Zestaw 3. Korzystanie z kolekcji

Poszczególne efekty uczenia się	Kryteria weryfikacji ich osiągnięcia
1. Korzysta z kontenerów sekwencyjnych	<ul style="list-style-type: none"> a) posługuje się kolekcjami: ArrayList oraz List, b) stosuje mechanizm sortowania obiektów w kolekcji, c) posługuje się metodami z klas ArrayList oraz List, d) charakteryzuje cechy kolekcji, w tym znaczenie iteratora, e) posługuje się wyrażeniami Lambda.
2. Korzysta z kontenerów asocjacyjnych	<ul style="list-style-type: none"> a) posługuje się kontenerami: set, map, b) stosuje iterator do zarządzania kontenerami asocjacyjnymi, c) posługuje się metodami dostępnymi w kontenerach asocjacyjnych.

Zestaw 4. Programowanie wielowątkowe, komunikacja sieciowa i korzystanie z baz danych

Poszczególne efekty uczenia się	Kryteria weryfikacji ich osiągnięcia
1. Korzysta z mechanizmu współbieżności	<ul style="list-style-type: none"> a) posługuje się interfejsem Runnable, b) stosuje mechanizm synchronizacji wątków, c) charakteryzuje problem zakleszczania wątków.

2. Korzysta z metod obsługi sieci	a) posługuje się klasami ServerSocket oraz Socket, b) stosuje metody klasy URLConnection.
3. Korzysta z baz danych za pomocą JDBC	a) posługuje się interfejsem JDBC, b) stosuje mechanizm nawiązywania połączenia z bazą danych, c) posługuje się zapytaniami SQL.

6. Wymagania dotyczące walidacji i podmiotów przeprowadzających walidację

1. Etap weryfikacji

1.1. Metody walidacji

Do weryfikacji efektów uczenia się stosuje się następujące metody:

- test teoretyczny,
- obserwacja w warunkach symulowanych (symulacja) lub rzeczywistych uzupełniona wywiadem swobodnym (rozmową z komisją).

Pozytywny wynik z części teoretycznej jest warunkiem przystąpienia do części praktycznej.

W szczególnych sytuacjach (np. sytuacja epidemiczna ograniczająca możliwość kontaktów bezpośrednich kandydata z komisją) możliwe jest zastosowanie innych metod walidacji lub jej form (np. zdalna) przy zachowaniu wszelkich zasad zapewniania jakości oraz obowiązku sprawdzenia wszystkich efektów uczenia się wraz z kryteriami weryfikacji.

1.2. Zasoby kadrowe

Weryfikację efektów kształcenia przeprowadza komisja składająca się z co najmniej 2 asesorów, w tym przewodniczącego komisji.

Przewodniczący komisji musi posiadać:

- wykształcenie wyższe informatyczne (co najmniej 7 poziom Polskiej Ramy Kwalifikacji),
- co najmniej 10 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela programowania.

Pozostali asesorzy muszą posiadać:

- wykształcenie wyższe informatyczne (co najmniej 6 poziom Polskiej Ramy Kwalifikacji),
- co najmniej 5 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela programowania.

1.3. Sposób organizacji walidacji oraz warunki organizacyjne i materialne

Instytucja prowadząca walidację zapewnia stanowisko komputerowe (jedno stanowisko dla jednego kandydata) wyposażone w:

- system operacyjny z interfejsem graficznym,
- połączenie z Internetem,
- przeglądarkę internetową,
- edytor tekstu lub środowisko programistyczne zapewniające możliwość tworzenia programów w języku Java (np. Eclipse),
- materiały biurowe (kartki, długopisy).

W przypadku zdalnego prowadzenia walidacji komisja zatwierdza warunki przystąpienia do walidacji w oparciu o warunki techniczne dające gwarancję samodzielnej realizacji walidacji przez kandydata, w szczególności możliwość stałej obserwacji kandydata z użyciem systemu teleinformatycznego zapewniającego wiarygodne sprawdzenie, czy osoba ubiegająca się o nadanie kwalifikacji rynkowej osiągnęła wyodrębnioną część lub całość efektów uczenia się wymaganych dla tej kwalifikacji. System teleinformatyczny i metody stosowane w walidacji muszą w szczególności umożliwiać identyfikację osoby przystępującej do walidacji, samodzielność pracy tej osoby i zabezpieczenie przebiegu walidacji przed ingerencją osób trzecich.

Sposób organizacji walidacji (w tym czas trwania oraz zastosowane narzędzia) musi umożliwić sprawdzenie posiadania wszystkich efektów uczenia się wymaganych dla niniejszej kwalifikacji.

Osoby walidowane powinny utworzyć program w języku Java według wskazanych założeń obejmujących podstawowe elementy języka, klasy i metody, kolekcje, wątki, obsługę sieci i baz danych. Możliwe jest korzystanie przez osoby uczestniczące w walidacji z materiałów dodatkowych w postaci literatury lub przykładowych fragmentów kodu znalezionych w sieci Internet.

2. Etapy identyfikowania i dokumentowania efektów uczenia się

Nie określa się wymagań dla tych etapów.

7. Warunki, jakie musi spełniać osoba przystępująca do walidacji

Nie dotyczy

8. Termin dokonywania przeglądu kwalifikacji

Nie rzadziej niż raz na 10 lat