MONITOR POLSKI

DZIENNIK URZĘDOWY RZECZYPOSPOLITEJ POLSKIEJ

Warszawa, dnia 21 grudnia 2022 r.

Poz. 1250

# OBWIESZCZENIE MINISTRA CYFRYZACJI[1)]

z dnia 7 grudnia 2022 r.

**w sprawie włączenia kwalifikacji rynkowej „Programowanie w języku C/C++” 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 C/C++” do Zintegrowanego Systemu Kwalifikacji.

Minister Cyfryzacji: wz. *J. Cieszyński*

1) Minister Cyfryzacji kieruje działem administracji rządowej – informatyzacja, na podstawie § 1 ust. 2 rozporządzenia Prezesa Rady Ministrów z dnia 6 października 2020 r. w sprawie szczegółowego zakresu działania Ministra Cyfryzacji (Dz. U. poz. 1716).

Załącznik do obwieszczenia Ministra Cyfryzacji
z dnia 7 grudnia 2022 r. (M.P. poz. 1250)

INFORMACJE O WŁĄCZENIU KWALIFIKACJI RYNKOWEJ „PROGRAMOWANIE W JĘZYKU C/C++” DO ZINTEGROWANEGO SYSTEMU KWALIFIKACJI

**1. Nazwa kwalifikacji rynkowej**

Programowanie w języku C/C++

**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 (ewentualnie odniesienie do poziomu Sektorowej Ramy Kwalifikacji)**

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 C/C++. W swojej pracy wykorzystuje zarówno mechanizm programowania proceduralnego, jak i obiektowego oraz korzysta z mechanizmu biblioteki STL. 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. Ponadto posługuje się wiedzą w zakresie paradygmatu obiektowego.

| **Zestaw 1. Programowanie proceduralne** | |
|---|---|
| **Poszczególne efekty uczenia się** | **Kryteria weryfikacji ich osiągnięcia** |
| Wykorzystuje środowiska programistyczne do tworzenia programów | – charakteryzuje zadania kompilatora i debuggera;<br>– analizuje błędy w kodzie za pomocą debuggera;<br>– charakteryzuje pojęcie biblioteki STL;<br>– kompiluje i uruchamia stworzone programy. |
| Posługuje się prostymi typami danych | – rozróżnia typy liczbowe całkowite i zmiennoprzecinkowe;<br>– rozpoznaje typ logiczny;<br>– rozróżnia typy znakowe i łańcuchowe;<br>– wykorzystuje typy liczbowe całkowite, zmiennoprzecinkowe, znakowe, typ logiczny i typ łańcuchowy. |
| Korzysta z operacji wejścia i wyjścia | – korzysta z różnych funkcji do operacji wejścia/wyjścia;<br>– posługuje się strumieniami do operacji wejścia/wyjścia. |
| Posługuje się instrukcjami sterującymi | – rozróżnia instrukcje warunkowe if oraz switch;<br>– rozróżnia pętle: for, while oraz do-while;<br>– korzysta z instrukcji if, for, break, continue. |

| | |
|---|---|
| Posługuje się dyrektywami preprocesora | – wykorzystuje dyrektywy preprocesora;<br>– definiuje różne rzeczy za pomocą dyrektywy #define;<br>– korzysta z dyrektyw preprocesora wpływających na przebieg kompilacji. |
| Posługuje się złożonymi typami danych | – rozróżnia rodzaje złożonych typów danych;<br>– korzysta z tablic jednowymiarowych i wielowymiarowych;<br>– wykorzystuje strukturę i unię;<br>– posługuje się typem wskaźnikowym i tablicami dynamicznymi;<br>– korzysta z typu wyliczeniowego. |
| Posługuje się operatorami | – rozróżnia operatory logiczne (&& – and, \|\| – or, ! – not);<br>– rozpoznaje operatory przesunięcia bitowego;<br>– wykorzystuje operatory logiczne, operatory przesunięcia bitowego oraz operatory bitowe AND, OR, XOR, NOT. |
| Stosuje własne funkcje | – rozróżnia różne sposoby przekazywania argumentów do funkcji;<br>– rozpoznaje przekazywanie parametrów do funkcji przez wartość, wskaźnik oraz referencję;<br>– tworzy własne funkcje;<br>– posługuje się rekurencyjnym wywołaniem funkcji. |
| Wykonuje operacje odczytywania i zapisywania plików | – rozróżnia pliki tekstowe oraz binarne;<br>– posługuje się typem plikowym;<br>– odczytuje i zapisuje dane. |

| **Zestaw 2. Programowanie obiektowe** | |
|---|---|
| **Poszczególne efekty uczenia się** | **Kryteria weryfikacji ich osiągnięcia** |
| Stosuje zasady programowania obiektowego | – charakteryzuje pojęcia: klasa, obiekt, metoda, pole, dziedziczenie, hermetyzacja, polimorfizm;<br>– dzieli zagadnienie na klasy;<br>– powołuje obiekty;<br>– projektuje aplikację z zastosowaniem hermetyzacji, dziedziczenia i polimorfizmu. |
| Korzysta z klas i obiektów | – definiuje pola klasy;<br>– określa zakres widoczności pól klasy i definiuje kwalifikatory dostępu;<br>– definiuje metody klasy;<br>– definiuje konstruktory, w tym konstruktor kopiujący, i destruktor klasy;<br>– definiuje listę inicjującą konstruktora;<br>– określa zakres widoczności metod klasy i definiuje kwalifikatory dostępu;<br>– deklaruje obiekty i odwołuje się obiektem do składowych klasy;<br>– definiuje składniki statyczne klasy;<br>– stosuje składnik statyczny klasy i metody do jego obsługi. |
| Korzysta z mechanizmu przyjaźni i przeciążonych operatorów | – tworzy funkcje zaprzyjaźnione z klasą;<br>– tworzy klasy zaprzyjaźnione;<br>– definiuje operatory dla klasy;<br>– posługuje się mechanizmem przyjaźni zarówno funkcji, jak i obiektów;<br>– posługuje się przeciążonymi operatorami arytmetycznymi, strumienia oraz nawiasów. |
| Definiuje klasy pochodne | – buduje hierarchię dziedziczenia klas w programie;<br>– wydziela metody i pola do odpowiednich klas w hierarchii dziedziczenia;<br>– definiuje klasy bazowe i pochodne;<br>– stosuje metody wirtualne, definiuje klasy abstrakcyjne. |

| | |
|---|---|
| Definiuje szablony klas i funkcji | – definiuje szablon funkcji z różnymi parametrami;<br>– określa szablon klasy;<br>– stosuje szablony funkcji oraz szablony klas. |
| Programuje obsługę wyjątków | – stosuje szkielet obsługi wyjątków z instrukcjami try i catch;<br>– stosuje instrukcję throw;<br>– opracowuje listę możliwych błędów wykonania aplikacji;<br>– definiuje obsługę dla błędów wykonania aplikacji w wyniku wykonywania różnych operacji. |

| **Zestaw 3. Wykorzystanie biblioteki STL** | |
|---|---|
| **Poszczególne efekty uczenia się** | **Kryteria weryfikacji ich osiągnięcia** |
| Korzysta z kontenerów sekwencyjnych | – posługuje się kolekcjami: vector, list oraz deque;<br>– stosuje mechanizm sortowania obiektów w kolekcji list;<br>– posługuje się metodami z klas vector, list oraz deque;<br>– charakteryzuje cechy kolekcji, w tym znaczenie iteratora. |
| Korzysta z kontenerów asocjacyjnych i adapterów | – posługuje się kontenerami: set, map oraz multitap;<br>– posługuje się kontenerem Stack oraz queue;<br>– stosuje do zarządzania kontenerami asocjacyjnymi iterator;<br>– posługuje się metodami dostępnymi w kontenerach asocjacyjnych i adapterach. |

## 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;

– obserwację w warunkach symulowanych (symulację) lub rzeczywistych uzupełnioną 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 przeprowadzającą weryfikację efektów kształcenia się, zwaną dalej „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 się przeprowadza komisja składająca się co najmniej z 2 osób.

Przewodniczący komisji musi posiadać:

– wykształcenie wyższe informatyczne (co najmniej 7 Poziom Polskiej Ramy Kwalifikacji);

– min. 10 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela informatyki lub programowania.

Pozostali członkowie komisji muszą posiadać:

– wykształcenie wyższe informatyczne (co najmniej 6 Poziom Polskiej Ramy Kwalifikacji);

– min. 5 lat udokumentowanego stażu pracy w zawodzie informatyka, programisty lub nauczyciela informatyki lub programowania.

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

Podmiot przeprowadzający walidację zapewnia:

1) stanowiska 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 C/C++ (np. Visual Studio);

2) 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 zatwierdza możliwość stałej obserwacji kandydata z użyciem systemu teleinformatycznego zapewniającego wiarygodne sprawdzenie, czy osoba ubiegająca się o nadanie kwalifikacji rynkowej osiągnęła wyodrębnioną część albo całość efektów uczenia się wymaganych dla tej kwalifikacji. System teleinformatyczny i metody stosowane w walidacji muszą w szczególności umożliwiać identyfikację kandydata przystępującego do walidacji, samodzielność pracy tego kandydata 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 C/C++ według wskazanych założeń obejmujących programowanie proceduralne, obiektowe i funkcje biblioteki STL. Możliwe jest korzystanie przez osoby walidowane z materiałów dodatkowych w postaci literatury lub przykładowych fragmentów kodu znalezionych w sieci Internet.

**2. Etap identyfikowania i dokumentowania**

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

## 7. Warunki, jakie musi spełniać kandydat przystępujący do walidacji

Nie dotyczy

## 8. Termin dokonywania przeglądu kwalifikacji

Nie rzadziej niż raz na 10 lat