

Wyposażenie dla ośrodków egzaminacyjnych na lata 2021-2023 r.

INF.04. Projektowanie, programowanie i testowanie aplikacji

Tabela 1. Powiązanie kwalifikacji z zawodami

Oznaczenie kwalifikacji	Nazwa kwalifikacji	Symbol cyfrowy zawodu	Nazwa zawodu
INF.04.	Projektowanie, programowanie i testowanie aplikacji	351406	Technik programista

Opis wyposażenia ośrodka egzaminacyjnego

- Miejsce egzaminowania** - pomieszczenie wyposażone w jednoosobowe stanowiska egzaminacyjne zapewniające samodzielne wykonanie zadania egzaminacyjnego, spełniające wymagania wynikające z przepisów bezpieczeństwa i higieny pracy, ochrony przeciwpożarowej oraz ochrony środowiska.

Tabela 2. Wyposażenie miejsca egzaminowania

Lp.	Wyposażenie dodatkowe i uzupełniające	Jednostka miary	Liczba
1.	Stolik i krzesła dla zespołu nadzorującego	szt.	w zależności od składu zespołu
2.	Stolik i krzesło dla obserwatora	szt.	1
3.	Tablica szkolna/plansza do zapisania czasu rozpoczęcia i zakończenia pracy zdających (kreda lub pisak)	szt.	1
4.	Zegar	szt.	1
5.	Apteczka	szt.	1
6.	Kosz na odpadki	szt.	1
7.	Długopis (zapasowy dla zdających)	szt.	wg potrzeb
8.	Identyfikator dla zdającego (oznaczony numerem stanowiska)	szt.	= liczbie zdających na zmianie
9.	Identyfikator dla zespołu nadzorującego (oznaczony wyłącznie napisem: PRZEWODNICZĄCY ZESPOŁU NADZORUJĄCEGO lub CZŁONEK ZESPOŁU NADZORUJĄCEGO)	szt.	dla każdej osoby
10.	Identyfikator dla obserwatora (oznaczony wyłącznie napisem: OBSERWATOR)	szt.	1
11.	Identyfikator dla administratora (opiekuna) pracowni (oznaczony wyłącznie napisem: ADMINISTRATOR)	szt.	1

2. Opis stanowiska egzaminacyjnego:

W skład stanowiska wchodzi stół wraz z zestawem komputerowym z systemem i oprogramowaniem niezbędnym do wykonania zadań egzaminacyjnych. Stanowisko powinno spełniać normy BHP dla stanowisk wyposażonych w komputery i urządzenia peryferyjne.

Tabela 3. Wyposażenie jednego stanowiska egzaminacyjnego:

l.p.	nazwa	Istotne funkcje- parametry techniczno-eksploatacyjne/ uwagi	Liczba
1	komputer osobisty	z zainstalowanym systemem operacyjnym oraz nagrywarką CD/DVD i zabezpieczony przed fizycznym dostępem do sieci komputerowej oraz Internetu (np. wyłączona/odinstalowana karta WiFi, odłączony patchcord od karty LAN) Uwaga! Nie jest dopuszczalne logiczne „odłączanie” komputerów od sieci, np. poprzez separację portów, tworzenie VLANów, itp. (dopuszcza się zastosowanie zewnętrznej nagrywarki CD/DVD oraz przenośnych komputerów osobistych)	1 szt.
Oprogramowanie			
3	system operacyjny	możliwy do zainstalowania i użytkowania na komputerze wymienionym w specyfikacji oraz umożliwiającym zainstalowanie oprogramowania wymienionego poniżej do wyboru: Windows, Linux, Mac OS	1 szt.
4	przeglądarka internetowa	np. Microsoft Edge, Google Chrome, Mozilla Firefox, Opera, Safari (w najnowszej wersji)	dowolna liczba
5	środowisko do tworzenia i uruchamiania programów obiektowych w konsoli systemowej	Wybrane środowisko programistyczne, zgodne z konfiguracją sprzętowo-systemową, ze wszystkimi niezbędnymi elementami do budowania konsolowych aplikacji obiektowych w jednym z języków: C++, C#, Java, Python.	1 szt.
6	środowisko do tworzenia i uruchamiania aplikacji desktopowych	Wybrane środowisko programistyczne, zgodne z konfiguracją sprzętowo-systemową, ze wszystkimi niezbędnymi elementami do budowania aplikacji desktopowych: - dla C++/C# – Visual Studio Community lub Xcode lub Qt - dla Python – pyCharm i PyQt - dla JAVA – IntelliJIDEA i Swing lub Qt Uwaga! Dopuszcza się użycie innego edytora programistycznego lub środowiska IDE.	1 kpl.

7	środowisko do tworzenia i uruchamiania aplikacji mobilnych	Środowisko programistyczne, zgodne z konfiguracją sprzętowo-systemową, ze wszystkimi niezbędnymi elementami do budowania aplikacji mobilnych dla systemów Android lub iOS lub Cross-Platform oraz wszystkie niezbędne elementy do emulacji programowej urządzenia mobilnego: Visual Studio Community, Android Studio, Xcode, Visual Studio for Mac	1 kpl.
8	środowisko programistyczne do tworzenia i uruchamiania aplikacji web po stronie klienta (front-end)	Środowisko programistyczne, zgodne z konfiguracją sprzętowo-systemową, ze wszystkimi niezbędnymi elementami do budowania aplikacji WEB po stronie klienta (front-end), wykorzystujące framework Angular lub bibliotekę React.js, wymagane elementy: - Node.js (jako środowisko instalacyjne – npm) - biblioteka Angular CLI wraz z językiem TypeScript lub biblioteka create-react-app - dowolne edytory programistyczne np. Visual Studio Code lub inne dowolnie wspomagające programowanie	1 kpl.
9	środowisko programistyczne do tworzenia i uruchamiania aplikacji web po stronie serwera (back-end)	Środowisko programistyczne wykorzystujące jedną z technologii: ASP.NET lub Django lub Node.js. Zgodne z konfiguracją sprzętowo-systemową, ze wszystkimi niezbędnymi elementami do budowania aplikacji web po stronie serwera (back-end) oraz współpracującym środowiskiem bazodanowym. Dowolny edytor z dowolnymi dodatkami wspomagającymi programowanie np. Visual Studio, Visual Studio Code - dla Node.js – środowisko Node.js, Node Package Manager, biblioteka framework Express, biblioteki do obsługi bazy danych MongoDB lub mySQL, oraz opcjonalnie inne dowolne biblioteki stosowane przez zdających np. underscore, jshint - dla Django – środowisko Python z wbudowaną bazą danych SQLite3 - dla ASP.NET – środowisko Visual Studio Community Edition umożliwiające tworzenie aplikacji we frameworku ASP.NET Core w tym w oparciu o MVC z wbudowaną bazą danych i Entity Framework, opcjonalnie dowolne dodatki do Visual Studio wspomagające programowanie (np. Productivity Power Tools, Web Essentials lub inne)	1 kpl.

10	oprogramowanie do archiwizacji	np. 7-Zip	1 kpl.
11	pakiet biurowy	Zgodny z konfiguracją sprzętowo-systemową np. Microsoft Office, Libre Office itp. wymagane składniki: edytor tekstu, arkusz kalkulacyjny	1 kpl
12	płyta CD/DVD i pisak do płyt	Zalecany jest nośnik dobrej jakości	1 kpl.

Tabela 4. Wyposażenie wspólne dla kilku stanowisk:

Lp.	Nazwa	Istotne funkcje-parametry techniczno-eksploatacyjne/uwagi/przykład	Maksymalna liczba zdających
komputery, peryferia			
1	Komputer do sprawdzenia jakości nagranej płyty	Komputer z zainstalowanym systemem operacyjnym umożliwiającym weryfikację poprawności nagrania płyty	10

Uwaga

Na 2 tygodnie przed egzaminem Ośrodek Egzaminacyjny otrzyma z OKE szczegółowe informacje dotyczące wyposażenia stanowisk egzaminacyjnych – **wskazania** do przygotowania stanowisk oraz specyfikację niezbędnych materiałów do wykonania zadania egzaminacyjnego (z przewidywaną kalkulacją kosztów), wg której powinien dokonać ich zakupu.

Dodatkowe instrukcje wspomagające konfigurację niektórych środowisk programistycznych

Szczegółowa instrukcja konfiguracji środowiska programistycznego

Poniżej przedstawiono przykład konfiguracji środowisk front-end i back-end, jest to przykład, z którego można skorzystać w celu prawidłowej konfiguracji środowiska. Według tych wytycznych zostaną skonfigurowane środowiska w ośrodkach sprawdzania. Konto Egzamin bez hasła. Konto z prawami administratora (wymagane np. przy wirtualnym środowisku Django)

Aplikacje Front-End

Do wyboru framework Angular lub biblioteka React

Angular

1. Zainstalować program Node.js. Dla macOS lub Windows pobrana najnowsza stabilna wersja z *nodejs.org*. Dla Linux (w przykładzie - Ubuntu) zainstalować za pomocą menedżera pakietów `sudo apt-get install nodejs npm`
2. Zainstalować bibliotekę Angular CLI oraz TypeScript. W linii komend:
 - a. `node --version` (sprawdzenie poprawności instalacji Node.JS)
 - b. `npm install -g @angular/cli` macOS i Linux - `sudo` (instalacja Angular za pomocą Node Package Manager)
UWAGA dotycząca Linux: w przypadku błędów, podać komendę `npm install -g --unsafe-perm --verbose @angular/cli`
 - c. `ng --version` (sprawdzenie poprawności zainstalowania Angular)
 - d. `npm install -g typescript` (instalacja języka TypeScript)
`tsc --version` (sprawdzenie poprawności instalacji TypeScript)
3. Zainstalować edytor Visual Studio Code (lub inny dowolny)
jeżeli zostanie wybrany VSCode dodać ścieżki do zmiennej środowiskowej PATH:
`<użytkownik egzamin>\AppData\Roaming\npm;`
`< użytkownik egzamin>\AppData\Local\Programs\Microsoft VS Code\bin`
lub za pomocą polecenia `code: VS Code -> shift + command + p` (macOS) lub `shift + control + p` (Windows) -> `code`
4. Sprawdzić czy działa środowisko poprzez utworzenie aplikacji Hello-World
 - a. na dysku C założyć katalog *Angular*
 - b. w katalogu *Angular* wydać polecenie `ng g new hello` (opcje: z routingiem, zwykłe CSS)
 - c. Skompilować za pomocą polecenia `ng serve`
 - d. Projekt jest dostępny w przeglądarce pod adresem `localhost:4200`. Ważne: projekt musi się znajdować na dysku C
 - e. Przed egzaminem nie należy usuwać przykładowej aplikacji, gdyż nowy projekt będzie tworzony offline
5. Utworzenie nowego projektu podczas egzaminu
 - a. Zdający może posłużyć się wcześniej utworzoną aplikacją Hello-World do rozwiązania zadania
 - b. W przypadku, gdy należy utworzyć nowy projekt na egzaminie (offline), kroki postępowania:
 - i. `ng new projekt1 --skip-install`

- ii. Skopiować folder `node_modules` z folderu aplikacji `hello-world` do folderu `projekt1`
- iii. W folderze `projekt1` komenda: `npm install`

Więcej informacji na ten temat na: <https://stackoverflow.com/questions/44127122/create-new-project-offline>

React

1. Instalacja Node.js – patrz instrukcja do Angular
2. Za pomocą Node Package Manager zainstalować pakiet `react` poprzez wydanie komendy:
`npm i -g create-react-app` macOS i Linux - `sudo`
3. Zainstalować edytor Visual Studio Code (lub inny dowolny) – patrz instrukcja do Angular
 - a. Jeżeli został zainstalowany VS Code zaleca się zainstalowanie dodatku ułatwiającego programowanie w React: `Simple React Snippets`
4. Sprawdzić czy działa środowisko poprzez utworzenie aplikacji Hello-World
 - a. Utworzyć folder `REACT` na dysku C
 - b. W folderze `REACT` wydać komendę: `create-react-app hello`
 - c. Przejść do katalogu `hello` (nazwa projektu)
 - d. Komenda: `npm start`
 - e. W wyniku komendy uruchomi się przeglądarka z adresem `localhost:3000` i wynikami działania aplikacji. Uwaga: projekt musi się znajdować na dysku C

Aplikacje Back-End

Do wyboru framework ASP.NET MVC lub Django lub Node.js

ASP.net

1. Zaleca się korzystanie z systemu Windows w przypadku wyboru ASP.NET.
2. Dla systemu Windows
 - a. Należy upewnić się czy w systemie jest zainstalowana najnowsza wersja .NET Core SDK.
 - b. Zainstalować środowisko ASP.NET: Visual Studio Community Edition umożliwiające tworzenie aplikacji we frameworku ASP.NET MVC
 - c. Zainstalować pakiet Entity Framework z poziomu Visual Studio (np. przez NuGet Package Manager)
 - d. Zainstalować bazę danych MS SQL Server lub skorzystać z bazy danych wbudowanej w Visual Studio (SQL Express)
 - e. Zainstalować dowolne elementy wspomagające programowanie
3. Dla MacOS i Linux
 - a. Zainstalować najnowszą wersję .NET Core SDK (np. ze strony <https://dotnet.microsoft.com/download>)
 - b. Zainstalować edytor programistyczny, np. Visual Studio Code oraz rozszerzenia: ASP.NET Helper, C#, C# Extensions, mssql
 - c. Zainstalować bazę danych SQL Server za pomocą narzędzia Docker z docker.com

Django

1. Zainstalować środowisko Python: `python.org` -> `downloads` -> najnowsza wersja; podczas instalacji dodać Python do zmiennej środowiskowej `PATH` (zaznaczyć `Add Python ... to PATH`)
Uwaga: W Linux jest bardzo prawdopodobne, że Python został zainstalowany wraz z systemem (należy się upewnić, próbując uruchomić interpreter Python). Zainstalować wpisując: `sudo apt install python3` (dla Ubuntu)
2. Utworzyć folder o nazwie projektu: `Projekt1`

3. `py -m venv Proj1` (aktywowane wirtualne środowisko z nazwą *Proj1*)
4. `Proj1\Scripts\activate` (Uruchomienie wirtualnego środowiska)
5. `py -m pip install --upgrade pip` (czy jest najnowsza wersja do zainstalowania Django?)
6. W folderze projektu (Projekt1) utworzyć plik *requirements.txt*, w którym zapisać wersję Django do zainstalowania, np.: `Django~=2.2.4`
7. `pip install -r requirements.txt` (instalacja Django)

Ważne uwagi do instalacji Django w systemach Linux i macOS

Przy niektórych poleceniach należy zastosować `sudo`. Zamiast polecenia `py`, polecenie `python` lub `python3`. Możliwe, że będzie wymagana w czasie zakładania wirtualnego środowiska instalacja modułu `python3-venv`. W niektórych wersjach Debian należy instalować wirtualne środowisko przez komendę `virtualenv`. Wirtualne środowisko uruchamia się za pomocą komendy `source myenv/bin/activate`. Jeżeli pojawi się błąd po użyciu polecenia `pip` należy zastosować polecenie `python -m pip install -U --force-reinstall pip`

Instrukcja utworzenia projektu (kropka jest częścią komendy):

1. `django-admin startproject nazwa-projektu .` (utworzenie projektu)
2. `py manage.py migrate` (utworzenie bazy danych)
3. `py manage.py runserver` (uruchomienie serwera django)
4. Serwer jest dostępny na localhost port 8000

Node.JS

1. Zainstalować program Node.js. Dla macOS lub Windows pobrana najnowsza stabilna wersja z nodejs.org. Dla Linux (Ubuntu) zainstalować za pomocą menedżera pakietów `sudo apt-get install nodejs npm`
2. Utworzyć nowy projekt i w nim utworzyć środowisko dla aplikacji WEB
 - a. `npm init` (utworzenie package.json) ew. z parametrem `--yes`
 - b. (opcjonalnie) Zainstalować wszelkie biblioteki wykorzystywane przez zdających na zajęciach i ułatwiające programowanie w node.js (np biblioteka `underscore` -> `npm install underscore`)
 - c. zainstalować bazę danych MongoDB lub MySQL (serwer i klient)
 - d. `npm i mongoose` lub `npm i mysql` (instalacja bibliotek do bazy danych)
 - e. `npm i express` (instalacja frameworka Express)

Aplikacje mobilne cross-platform przy wykorzystaniu biblioteki Xamarin (tylko Windows i macOS)

Dla systemu Windows: wymagany system Windows 10, Microsoft Visual Studio Community Edition, najnowsza stabilna wersja, Android Device Manager z przynajmniej jednym wirtualnym urządzeniem

Dla systemu macOS: Xamarin Studio, najnowsza stabilna wersja lub Visual Studio 2019 for Mac

Instrukcje instalacji

- Dla Windows:
 - W MS Visual Studio zainstalować wszystkie niezbędne elementy do tworzenia aplikacji cross-platform w oparciu o język C#, sekcja *Mobile development with .NET*
 - Skonfigurować emulator systemu Android za pomocą Android Device Manager (dodać urządzenie wirtualne)

- Sprawdzić poprawność instalacji i emulacji tworząc projekt w sekcji Visual C# -> Cross-Platform -> Mobile App i uruchamiając go na platformie Android
- Dla macOS:
 - Zainstalować Xamarin Studio lub Visual Studio 2019 for Mac z elementami *Android + Xamarin.Forms* oraz *iOS + Xamarin.Forms, macOS, .NET Core, Xamarin Workbooks & Inspector*
 - Sprawdzić poprawność instalacji i emulacji tworząc projekt w sekcji Multiplatform -> Xamarin.Forms -> Forms App i uruchamiając go na platformie iPhone

Aplikacje okienkowe dla Windows przy wykorzystaniu WPF (tylko Windows)

Jedynie dla systemu Windows: Microsoft Visual Studio Community Edition, najnowsza stabilna wersja

Instrukcje instalacji

- dla Windows:
 - Podczas instalacji Visual Studio wybrać *.NET desktop development*

Sprawdzić poprawność instalacji tworząc i uruchamiając projekt w sekcji Visual C# -> Windows Desktop -> WPF App