

Nazwa kwalifikacji: **Projektowanie, programowanie i testowanie aplikacji**
Oznaczenie kwalifikacji: **INF.04**
Numer zadania: **03**
Wersja arkusza: **SG**

Wypełnia zdający

Numer PESEL zdającego*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Miejsce na naklejkę z numerem
PESEL i z kodem ośrodka

Czas trwania egzaminu: **180** minut.

INF.04-03-23.06-SG

EGZAMIN ZAWODOWY

Rok 2023

CZĘŚĆ PRAKTYCZNA

**PODSTAWA PROGRAMOWA
2019**

Instrukcja dla zdającego

1. Na pierwszej stronie arkusza egzaminacyjnego wpisz w oznaczonym miejscu swój numer PESEL i naklej naklejkę z numerem PESEL i z kodem ośrodka.
2. Na KARCIE OCENY w oznaczonym miejscu przyklej naklejkę z numerem PESEL oraz wpisz:
 - swój numer PESEL*,
 - oznaczenie kwalifikacji,
 - numer zadania,
 - numer stanowiska.
3. Sprawdź, czy arkusz egzaminacyjny zawiera 5 stron i nie zawiera błędów. Ewentualny brak stron lub inne usterki zgłoś przez podniesienie ręki przewodniczącemu zespołu nadzorującego.
4. Zapoznaj się z treścią zadania oraz stanowiskiem egzaminacyjnym. Masz na to 10 minut. Czas ten nie jest wliczony do czasu trwania egzaminu.
5. Czas rozpoczęcia i zakończenia pracy zapisze w widocznym miejscu przewodniczący zespołu nadzorującego.
6. Wykonaj samodzielnie zadanie egzaminacyjne. Przestrzegaj zasad bezpieczeństwa i organizacji pracy.
7. Po zakończeniu wykonania zadania pozostaw arkusz egzaminacyjny z rezultatami oraz KARTĘ OCENY na swoim stanowisku lub w miejscu wskazanym przez przewodniczącego zespołu nadzorującego.
8. Po uzyskaniu zgody zespołu nadzorującego możesz opuścić salę/miejsce przeprowadzania egzaminu.

Powodzenia!

* w przypadku braku numeru PESEL – seria i numer paszportu lub innego dokumentu potwierdzającego tożsamość

Zadanie egzaminacyjne

UWAGA: katalog z rezultatami pracy oraz płytę należy opisać numerem zdającego, którym został podpisany arkusz, czyli numerem PESEL lub w przypadku jego braku numerem paszportu. Dalej w zadaniu numer ten jest nazwany numerem zdającego.

Wykonaj aplikację konsolową oraz webową według wskazań. Wykonaj dokumentację do aplikacji konsolowej, zgodnie z opisem w części III instrukcji do zadania. Wykorzystaj konto **Egzamin** bez hasła.

Utwórz folder i nazwij go numerem zdającego. W folderze utwórz podfoldery: *konsola*, *web*, *dokumentacja*. Po wykonaniu każdej aplikacji, jej pełny kod (cały folder projektu) spakuj do archiwum. Następnie pozostaw w podfolderze jedynie pliki źródłowe, których treść była modyfikowana, plik wykonywalny, jeśli jest to możliwe oraz spakowane archiwum. Dokumentację aplikacji konsolowej oraz zrzuty ekranu dokumentujące uruchomienie obu aplikacji umieść w folderze *dokumentacja*.

Część I. Aplikacja konsolowa

Korzystając z mechanizmów programowania obiektowego zaprojektuj część logiki systemu wirtualnej wypożyczalni filmów. Zaimplementuj aplikację zawierającą klasę o nazwie *film* z metodami *get*, *set* oraz *inkrementuj*.

Założenia klasy:

- Obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym, jeden z: C++, C#, Java, Python.
- Klasa *film* zawiera dwa pola zawierające tytuł oraz liczbę wypożyczeń. Dla uproszczenia można przyjąć, że tytuł może mieć maksymalnie 20 znaków. Dostęp do obu pól ma jedynie klasa. W przypadku późniejszego rozszerzenia klasy, klasy potomne powinny również mieć dostęp do tych pól.
- Klasa *film* zawiera funkcjonalność:
 - inicjalizację pól wartością pustą lub 0 w zależności od typu
 - metodę ustawiającą tytuł (*ang. setter*)
 - metodę pobierającą tytuł (*ang. getter*)
 - metodę pobierającą liczbę wypożyczeń (*ang. getter*)
 - bezargumentową metodę realizującą inkrementację pola przechowującego liczbę wypożyczeń. Metoda nie zwraca żadnej wartości

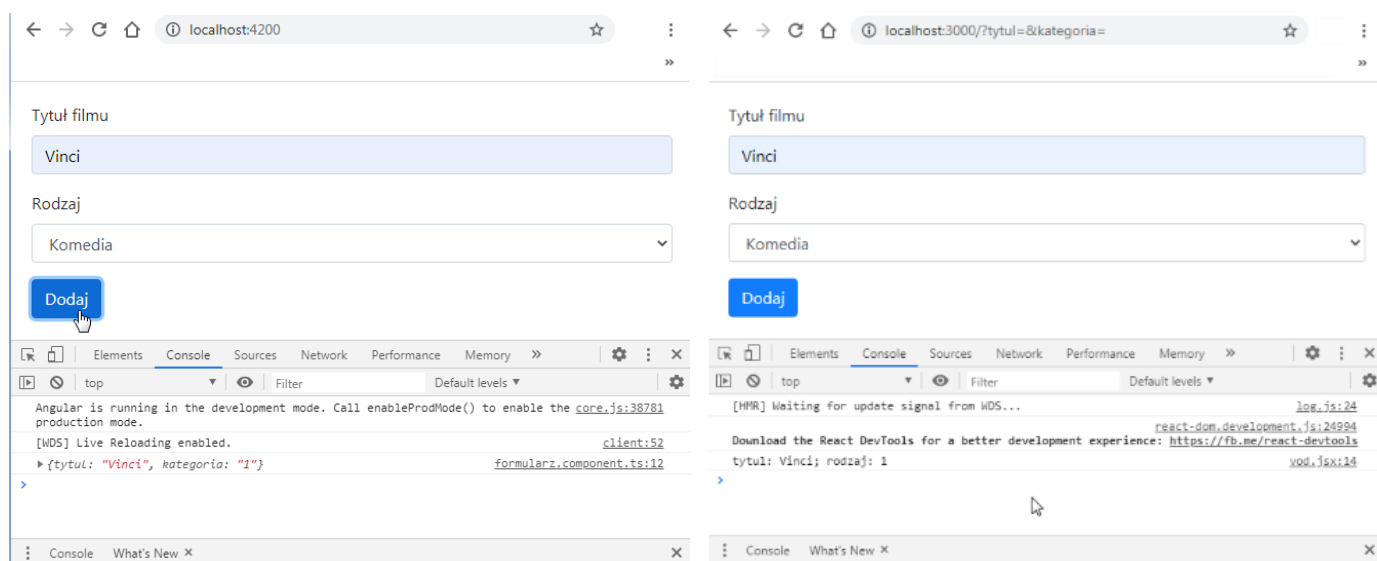
Założenia aplikacji:

- Inicjalizacja obiektu i wyświetlenie zawartości jego pól
- Test działania metody *set*, poprzez ustawienie dowolnej wartości dla pola obiektu,
- Test działania metody *get*, poprzez pobranie wartości z pola obiektu i wyświetlenie jej na ekranie
- Test metody inkrementacji pola przechowującego liczbę wypożyczeń zrealizowany w ten sposób, że wyświetlona jest wartość pola przed i po zastosowaniu metody
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy pól i metod
- Dokumentacja do aplikacji wykonana zgodnie z wytycznymi z części III zadania egzaminacyjnego.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *konsola* powinno znaleźć się archiwum całego projektu o nazwie *konsola.zip*, plik z kodem źródłowym programu oraz plik wykonywalny, jeżeli istnieje

Część II. Aplikacja Web

Wykonaj aplikację internetową typu front-end obsługującą formularz z zastosowaniem dostępnego na stanowisku egzaminacyjnym frameworka Angular lub biblioteki React. Zastosuj bibliotekę Bootstrap do zdefiniowania stylu formularza.



Obraz 1a. Formularz wykonany w frameworku Angular. Wybrano przycisk Dodaj

Obraz 1b. Formularz wykonany z biblioteką React.js. Wybrano przycisk Dodaj

Na obrazach 1a i 1b przedstawiono działanie formularza przygotowanego w środowisku Angular i React.js po wybraniu przycisku Dodaj. W konsoli widoczne wyświetlenie danych wprowadzonych do formularza

Założenia aplikacji

- Aplikacja składa się z jednego komponentu formularza.
- Formularz składa się z:
 - pola edycyjnego i jego etykiety o treści „Tytuł filmu”
 - listy rozwijalnej i jej etykiety o treści „Rodzaj”
 - przycisku „Dodaj”
- Lista rozwijalna ma 5 opcji:
 - pustą
 - Komedia o wartości 1
 - Obyczajowy o wartości 2
 - Sensacyjny o wartości 3
 - Horror o wartości 4
- Aplikacja w stanie początkowym wyświetla puste pola formularza
- Po wybraniu przycisku „Dodaj” jest generowane zdarzenie, które ma za zadanie wypisanie danych z formularza w konsoli przeglądarki (w miejscu trzech kropek wartości pól formularza)
 - w Angular postaci: {tytul: "...", kategoria: "..."} – obraz 1a
 - w React.js postaci: tytul: ...; rodzaj: ... – obraz 1b
- Strona jest formatowana stylem: `body { padding: 20px; }`
- Elementy formularza są formatowane stylami biblioteki Bootstrap. Szablon HTML formularza należy zbudować sugerując się pomocą zamieszczoną w Tabeli 1. Należy zastosować znaczące nazwy dla identyfikatorów pól formularza
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.
- Dokumentacja do programu wykonana zgodnie z wytycznymi z części III zadania egzaminacyjnego.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *web* powinno znaleźć się archiwum całego projektu o nazwie *web.zip* oraz pliki z kodem źródłowym które były modyfikowane w czasie egzaminu.

Część III. Dokumentacja utworzonych aplikacji

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej utwórz nagłówek klasy, według listingu 1. Nagłówek powinien znaleźć się w kodzie źródłowym nad klasą. W miejscu nawiasów <> należy podać nazwę klasy, nazwy pól oraz związane informacje (kilka słów) – zgodnie ze wzorcem. W miejscu informacji należy podać opis, co reprezentuje klasa. W miejscu autora należy podać numerem zdającego.

UWAGA: Dokumentację umieścić w komentarzu (wieloliniowym lub kilku jednoliniowych). Znajdujący się w listingu 1 wzór dokumentacji jest bez znaków początku i końca komentarza, gdyż te są różne dla różnych języków programowania

Listing 1. Wzór dokumentacji klasy

```
*****
nazwa klasy: <tu wstaw nazwę klasy>
pola:        <nazwa pola> - <co przechowuje>
             <nazwa pola> - <co przechowuje>
metody:      <nazwa metody>, <co zwraca> - <opis>
informacje: <opis>
autor:       <numer zdającego>
*****
```

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji utworzonych podczas egzaminu. Zrzuty powinny obejmować cały obszar ekranu monitora z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na zrzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem lub okno terminala z kompilacją projektu. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na zrzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym. Wymagane zrzuty ekranu:

- dowolna liczba zrzutów ekranowych z aplikacji konsolowej nazwane *konsola1*, *konsola2* itd.
- dowolna liczba zrzutów ekranowych z aplikacji webowej nazwane *web1*, *web2* itd.

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać podpisane zrzuty ekranu oraz zapisane informacje:

- nazwę systemu operacyjnego, na którym pracował zdający,
- nazwy środowisk programistycznych, z których zdający korzystał na egzaminie,
- nazwy języków programowania / frameworków / bibliotek użytych podczas tworzenia aplikacji,
- opcjonalnie komentarz do wykonanej pracy.

Zrzuty ekranu i dokument umieść w folderze o nazwie *dokumentacja*.

UWAGA: Nagraj płytę z rezultatami pracy. W folderze z numerem zdającego powinny się znajdować podfoldery *dokumentacja*, *konsola*, *web*. W folderze *dokumentacja* pliki: *konsola.jpg*, *web.jpg*, *egzamin*. W folderze *konsola*: cały projekt aplikacji konsolowej. W folderze *web*: cały projekt aplikacji webowej, ewentualnie inne przygotowane pliki. Po nagraniu płyty sprawdź poprawność nagrania. Opisz płytę numerem zdającego i pozostaw na stanowisku, zapakowaną w pudełku wraz z arkuszem egzaminacyjnym.

Czas przeznaczony na wykonanie zadania wynosi 180 minut.

Ocenie będą podlegać 4 rezultaty

- implementacja, kompilacja, uruchomienie programu,
- aplikacja konsolowa,
- aplikacja web,
- dokumentacja aplikacji.

Tabela 1. Wybrane elementy frameworka Angular, biblioteki React.js i biblioteki Bootstrap

<p>Angular</p> <p>To use ngModel i ngForm add: <code>import { FormsModule } from '@angular/forms';</code> in <code>app.module.ts</code> file. Add <code>FormsModule</code> to <code>imports</code> table</p> <p>To use Bootstrap add to <code>styles.css</code>: <code>@import "~bootstrap/dist/css/bootstrap.css";</code></p>
<p>React.js</p> <p>To use Bootstrap add: <code>import 'bootstrap/dist/css/bootstrap.css';</code></p>
<p>Bootstrap Forms (Źródło https://getbootstrap.com/docs/4.0/components/forms/)</p> <p>Be sure to use an appropriate type attribute on all inputs (e.g., email for email address or number for numerical information) to take advantage of newer input controls like email verification, number selection, and more.</p> <p>Textual form controls—like <code><input></code>s, <code><select></code>s, and <code><textarea></code>s—are styled with the <code>.form-control</code> class. Included are styles for general appearance, focus state, sizing, and more.</p> <pre><form> <div class="form-group"> <label for="exampleInputEmail1">Email address</label> <input type="email" class="form-control" id="exampleInputEmail1" /> </div> <div class="form-group"> ... </div> </form></pre> <p>Important! In React render method use <code>className</code> instead of <code>class</code>; <code>htmlFor</code> instead of <code>for</code>.</p>
<p>Bootstrap buttons (Źródło https://getbootstrap.com/docs/4.0/components/buttons/)</p> <p>Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control. The <code>btn</code> classes are designed to be used with the <code><button></code> element. Add modifier classes as: <code>btn-primary</code>, <code>btn-secondary</code>, <code>btn-success</code> and more to <code>btn</code> class in order to add background colors. e.g. <code><button type="button" class="btn btn-success">Success</button></code></p>

