CENTRALNA KOMISJA EGZAMINACYJNA

Arkusz zawiera informacje prawnie chronione do momentu rozpoczęcia egzaminu

Nazwa kwalifikacji: **Projektowanie, programowanie i testowanie aplikacji**
Symbol kwalifikacji: **INF.04**
Numer zadania: **01**
Wersja arkusza: **SG**

Wypełnia zdający

Numer PESEL zdającego*

Miejsce na naklejkę z numerem PESEL i z kodem ośrodka

Czas trwania egzaminu: **180** minut.

INF.04-01-25.01-SG

EGZAMIN ZAWODOWY

Rok 2025

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 8 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 wliczany 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: numer, którym został podpisany arkusz egzaminacyjny (PESEL lub w przypadku jego braku numer paszportu) jest w zadaniu nazywany ***numerem zdającego****.*

Wykonaj aplikację konsolową oraz webową według wskazań. Wykonaj dokumentację 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: *konsolowa*, *webowa*, *dokumentacja*. Po wykonaniu każdej aplikacji, jej pełny kod (cały folder projektu) **spakuj do archiwum**. Następnie pozostaw w podfolderze jedynie spakowane archiwum, pliki źródłowe, których treść była modyfikowana oraz jeśli jest to możliwe plik wykonywalny.

Część I. Aplikacja konsolowa

Za pomocą narzędzi do tworzenia aplikacji konsolowych zaimplementuj program realizujący różne operacje na tablicach.

Założenia aplikacji:

- Zastosowany obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C#, lub Java, lub Python
- Tablica oraz operacje na niej wykonywane są implementowane z wykorzystaniem klasy
- Pola klasy:
  - Tablica liczb całkowitych (ma być tradycyjną tablicą, a w Python listą)
  - Liczba elementów tablicy zapisana jako liczba całkowita. Pole przechowuje faktyczną liczbę elementów. Wszystkie operacje są ograniczone wartością tego pola
  - Oba pola są dostępne tylko w tej klasie oraz niedostępne dla klas potomnych
- Konstruktor klasy:
  - Przyjmuje jako argument rozmiar tablicy
  - Ustawia wartość pola liczby elementów tablicy na wartość argumentu
  - Wypełnia tablicę, będącą polem klasy, pseudolosowymi liczbami całkowitymi z zakresu od 1 do 1000
- Metody klasy:
  - Wyświetlająca wszystkie elementy tablicy w postaci „<index_tablicy>: <wartość>”. Nie zwraca wartości
  - Wyszukująca pierwsze wystąpienie wartości, przekazanej jako argument. Metoda zwraca indeks szukanego elementu lub liczbę -1, gdy elementu nie znaleziono
  - Wyświetlająca wszystkie wartości nieparzyste z tablicy i zwracająca ich liczbę
  - Licząca średnią arytmetyczną wartości w tablicy i zwracająca tą wartość
  - Wszystkie metody są dostępne poza klasą
- Program główny: (fragment działania jest widoczny na obrazie 1)
  - Tworzy obiekt klasy z rozmiarem tablicy większym od 20

```
44: 406
45: 81
46: 613
47: 267
48: 196
49: 588
Liczby nieparzyste:
559
515
999
203
655
887
959
591
999
345
875
915
737
911
971
275
109
253
669
957
325
931
81
613
267
Razem nieparzystych: 25
Średnia wszystkich elemenetów: 525
```

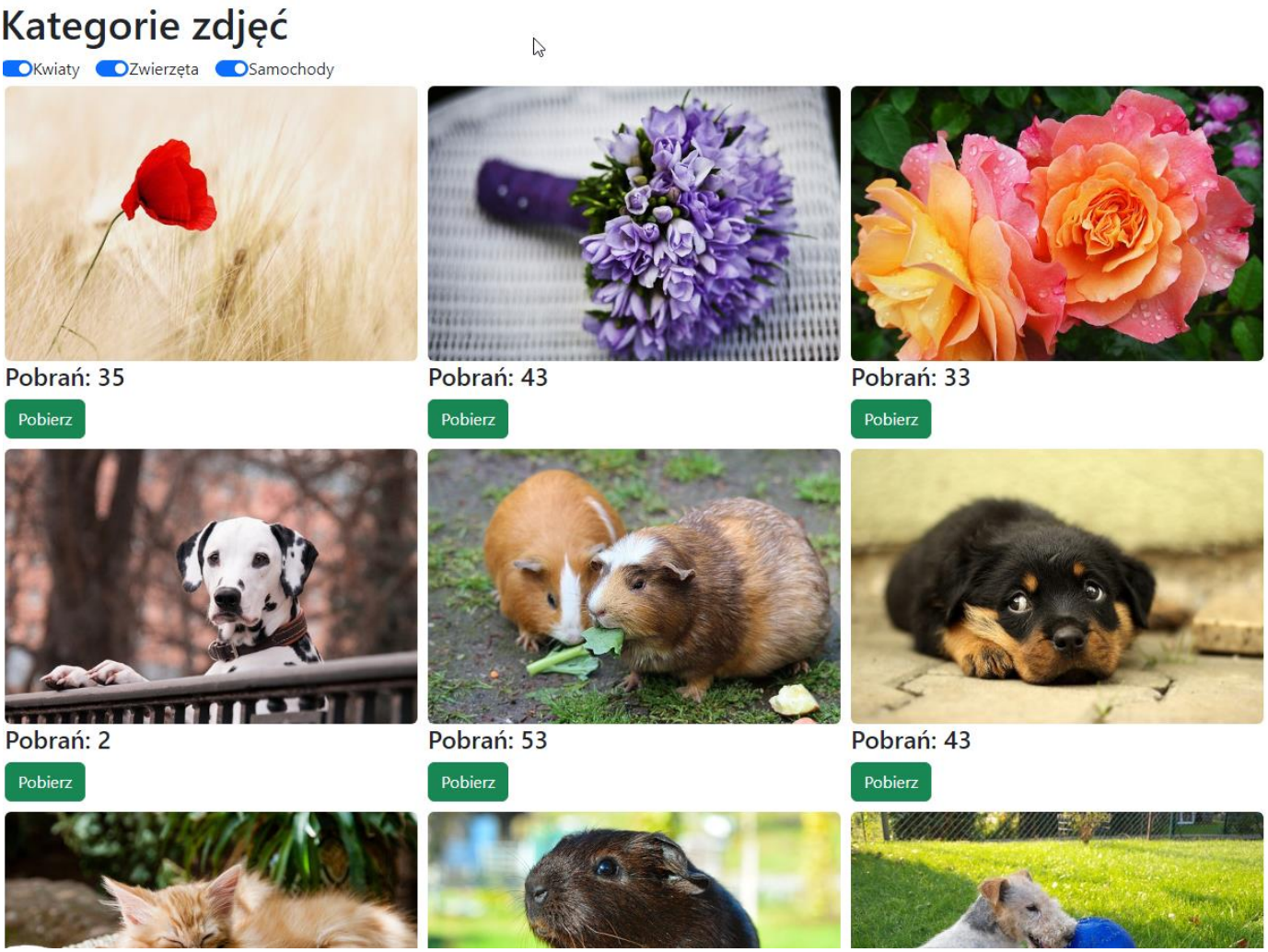
Obraz 1. Fragment działania aplikacji konsolowej

  - Sprawdza działanie wszystkich metod:
    - Wyświetlającej wszystkie elementy tablicy
    - Wyszukującej: jeżeli wartość została wyszukana, wyświetlany jest w programie głównym komunikat z wartością indeksu wyszukanej. W przeciwnym wypadku nic nie jest wyświetlane (taka sytuacja ma miejsce na obrazie 1)
    - Wyświetlającej liczby nieparzyste oraz wyświetlana jest ich ilość z odpowiednim komentarzem
    - Liczącej średnią, po czym w programie głównym wyświetlana jest wartość średniej
- Komunikacja z użytkownikiem musi być zrozumiała.
- Program powinien być zapisany czytelnie, z zachowaniem zasad czystego formatowania kodu
- Dla obiektów, pól, metod i zmiennych należy stosować znaczące nazewnictwo angielskie lub polskie. Dopuszcza się dla klasy i pola tablicowego nazewnictwo ogólne (np. tab, tablica, table)
- Do kodu należy dołączyć dokumentację, która została opisana w części III zadania egzaminacyjnego.

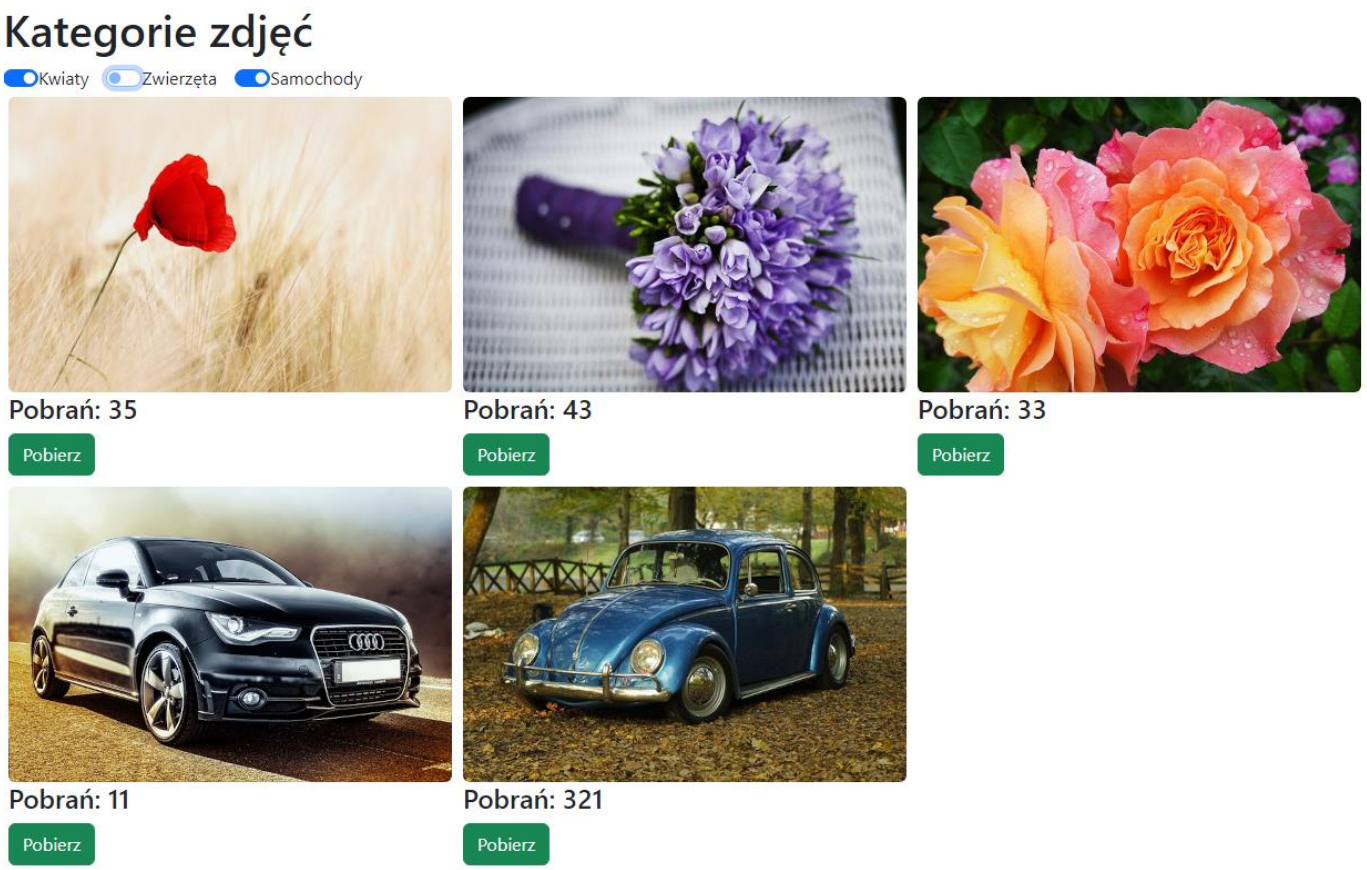
Kod aplikacji przygotuj do nagrania na płytę. W folderze *konsolowa* powinno znaleźć się archiwum całego projektu o nazwie *konsola.zip*, plik z kodem źródłowym programu oraz plik wykonywalny, jeżeli istnieje.

Część II. Aplikacja webowa

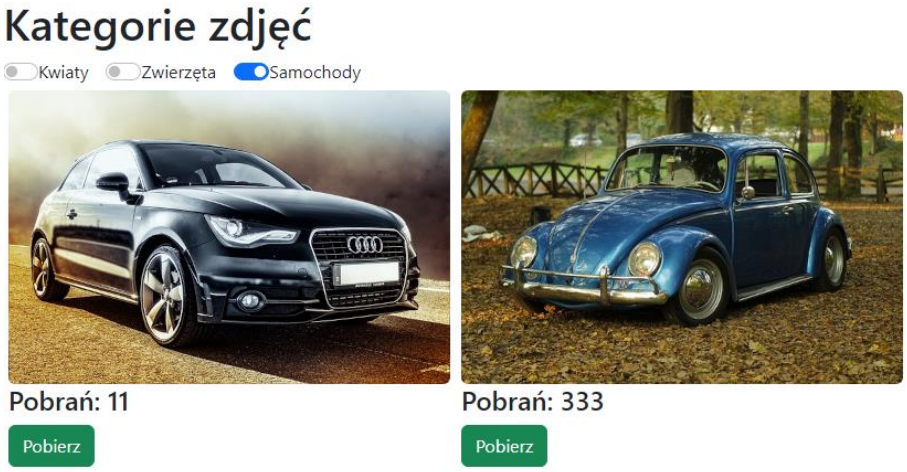
Z zastosowaniem dostępnego na stanowisku egzaminacyjnym frameworka Angular lub biblioteki React.js wykonaj aplikację internetową typu front-end realizującą funkcję kategoryzacji zdjęć w galerii. Na obrazach 2, 3, 4 przedstawiono działanie aplikacji. W zależności od zastosowanego narzędzia wygląd aplikacji może nieznacznie się różnić. Na pulpicie znajduje się archiwum, z materiałami do wykonania zadania, o nazwie *pliki3.zip* zabezpieczone hasłem: **K@tegorie)**



Obraz 2. Aplikacja w stanie początkowym



Obraz 3. Działanie aplikacji: odznaczono „Zwierzęta”, zostały obrazy kwiatów i samochodów



Obraz 4. Wielokrotnie wciśnięto przycisk drugiego zdjęcia. Liczba pobrań w porównaniu z obrazem 3 wzrosła

Założenia aplikacji

- Aplikacja składa się z jednego komponentu, którego widok w stanie początkowym zaprezentowany jest na obrazie 2 (zdjęcia mogą być wyświetlane w dowolnej kolejności)
- Do utworzenia aplikacji należy wykorzystać zdjęcia oraz plik *dane.txt* wypakowane z archiwum
- Obrazy należy umieścić w folderze *assets* (*egzamin/src/assets* lub *egzamin/public/assets*)
- Dokument *dane.txt* zawiera listę obiektów zdjęć, którą należy skopiować jako elementy tablicy. Każdy obiekt zdjęcia zawiera pola:
  - id
  - alt (tekst alternatywny dla zdjęcia)
  - filename (nazwa pliku ze zdjęciem)
  - category (1 dla kategorii kwiaty, 2 zwierzęta, 3 samochody)
  - downloads (liczba pobrań zdjęcia)
- Komponent składa się z:
  - Nagłówka pierwszego stopnia o treści: „Kategorie zdjęć”
  - Trzech pól switch (checkbox) domyślnie włączonych, o etykietach: Kwiaty, Zwierzęta, Samochody
  - Bloków zdjęć, które są wyświetlane warunkowo, w zależności od ustawień pól switch. Bloki są wyświetlone jeden obok drugiego, zawierają zdjęcie, nagłówek 4 stopnia z liczbą pobrań oraz przycisk o treści „Pobierz”. Układ elementów jest przedstawiony na obrazie 2
    - Zdjęcia są formatowane stylem: marginesy zewnętrzne 5 px, zaokrąglone rogi
- Przyciski oraz pola switch są stylowane zgodnie z przykładami w tabeli 1
- W stanie początkowym włączone są wszystkie pola switch co powoduje wyświetlenie wszystkich zdjęć
- Wyświetlane są tylko zdjęcia z kategorii dla której jest włączone pole switch (obraz 4)
- Gdy przycisk „Pobierz” zostanie kliknięty, wzrasta o jeden liczba pobrań dla danego zdjęcia. Liczba pobrań jest zapisywana w tablicy z obiektami zdjęć, co na bieżąco powoduje wyświetlenie tej modyfikacji na ekranie (obraz 5)
- W aplikacji zastosowano pętle oraz warunki do wyświetlenia bloków zdjęć. Aplikacja jest napisana uniwersalnie i działa poprawnie też dla innej liczby zdjęć
- Aplikacja powinna być zapisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podejmij próbę uruchomienia aplikacji w przeglądarce. Informacje dotyczące zrzutów ekranu znajdują się w części III zadania egzaminacyjnego.

Kod aplikacji przygotuj do nagrania na płytę. W folderze *webowa* powinno znaleźć się archiwum całego folderu projektu o nazwie *web.zip* oraz pliki z kodem źródłowym, które były modyfikowane przez zdającego.

Część III. Dokumentacja utworzonych aplikacji

Wykonaj dokumentację do aplikacji utworzonych na egzaminie. W kodzie źródłowym aplikacji konsolowej za pomocą komentarza utwórz nagłówek dowolnej metody, według wzoru z listingu 1. Komentarz powinien znaleźć się nad lub pod nazwą metody. W miejscu nawiasów <> należy podać odpowiednie opisy. W sekcji parametry należy umieścić opis wszystkich argumentów metody lub zapisać „brak” w przypadku metody bezparametrowej.

UWAGA: Dokumentację należy 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 metody (liczba gwiazdek dowolna)

```
*******************************************
nazwa metody:          <nazwa>
opis metody:           <krótki opis, co robi metoda>
parametry:             <nazwa i opis parametru1, lub „brak”>
                       <nazwa i opis parametru2>
                       ...
zwracany typ i opis:   <nazwa typu i opis co jest zwracane lub „brak”>
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. Wykonać należy tyle zrzutów, ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

- Aplikacja konsolowa – dowolna liczba zrzutów nazwanych *konsola1*, *konsola2,* ...
- Aplikacja webowa – dowolna liczba zrzutów nazwanych *web1*, *web2,* ... (stan początkowy, różne kombinacje dla pól switch, działanie przycisku „Pobierz”)

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać zapisane informacje o wykorzystanych w czasie egzaminu narzędziach:

- Nazwę systemu operacyjnego
- Nazwy środowisk programistycznych
- Nazwy języków programowania

Zrzuty ekranu i dokument umieść w podfolderze *dokumentacja*.

UWAGA: Nagraj płytę z rezultatami pracy. W folderze z numerem zdającego powinny się znajdować podfoldery dokumentacja, konsolowa, webowa. W folderze dokumentacja: pliki ze zrzutami oraz plik egzamin. W folderze konsolowa: spakowany cały projekt aplikacji konsolowej, źródło, opcjonalnie plik wykonywalny. W folderze webowa: spakowany cały projekt aplikacji web, pliki ze źródłami komponentu, html, css i inne modyfikowane pliki. 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 biblioteki Bootstrap

Angular

To use Bootstrap add to *styles.css*: @import "~bootstrap/dist/css/bootstrap.css";

React.js

To use Bootstrap add: import 'bootstrap/dist/css/bootstrap.css';

Bootstrap switch

A switch has the markup of a custom checkbox but uses the .form-switch class to render a toggle switch. Switches also support the disabled attribute. Examples of unchecked and checked switch:

```
<div class="form-check form-switch">
  <input class="form-check-input" type="checkbox" id="ctrId">
  <label class="form-check-label" for="ctrId">My label</label>
</div>
<div class="form-check form-switch">
  <input class="form-check-input" type="checkbox" id=" ctrId2">
  <label class="form-check-label" for=" ctrId2">My label</label>
</div>
```

To Inline switches group checkboxes or radios on the same horizontal row by adding .form-check-inline to any .form-check.

Bootstrap buttons

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control. The btn classes are designed to be used with the <button> element. Add class modifier as: btn-primary, btn-secondary, btn-success, btn-danger, btn-warning, btn-info, btn-light, btn-dark to btn class in order to add background colors.
e.g.

```
<button type="button" class="btn btn-success">Success</button>
```

Wypełnia zdający

Do arkusza egzaminacyjnego dołączam płytę CD opisaną numerem PESEL | | | | | | | | | | | |**,**
której jakość nagrania została przeze mnie sprawdzona.

Wypełnia Przewodniczący ZN

Potwierdzam, że do arkusza egzaminacyjnego dołączona jest płyta CD, opisana numerem PESEL zdającego.

……
Czytelny podpis Przewodniczącego ZN