

Nazwa kwalifikacji: **Projektowanie, programowanie i testowanie aplikacji**Symbol kwalifikacji: **INF.04**Numer zadania: **02**Wersja arkusza: **SG**

Wypełnia zdający

Numer PESEL zdającego*

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

Miejsce na naklejkę z numerem
PESEL i z kodem ośrodkaCzas trwania egzaminu: **180** minut.

INF.04-02-25.06-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 7 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 desktopową i testy jednostkowe dla aplikacji konsolowej według wskazań. Wykorzystaj konto **Egzamin** bez hasła.

Utwórz folder i nazwij go numerem zdającego. W folderze utwórz podfoldery: *konsolowa*, *desktopowa*, *testy*. Po wykonaniu każdej aplikacji, jej pełny kod (cały folder projektu), **spakuj do archiwum**. Następnie pozostaw w podfolderze jedynie skopiowane z projektu pliki źródłowe, których treść była modyfikowana, plik wykonywalny, jeśli jest to możliwe, oraz spakowane archiwum.

Część I. Aplikacja konsolowa

Za pomocą narzędzi do tworzenia aplikacji konsolowych utwórz program implementujący algorytm szyfru Cezara.

Założenia aplikacji:

- Zastosowany obiektowy język programowania zgodny z zainstalowanym na stanowisku egzaminacyjnym: C++ lub C#, lub Java, lub Python
- Podejście obiektowe lub strukturalne
- Szyfrowanie odbywa się w osobnej metodzie lub funkcji, która zawiera:
 - Dwa argumenty: tekst jawny oraz klucz (przy podejściu obiektowym dopuszcza się bez parametrów)
 - Wartość zwracaną: tekst zaszyfrowany
 - Implementację algorytmu szyfrowania kodem Cezara
- W programie należy stosować znaczące, angielskie lub polskie nazewnictwo
- Program powinien być zapisany czytelnie, z zachowaniem zasad czystego formatowania kodu
- Do kodu należy dołączyć testy jednostkowe opisane w części III zadania egzaminacyjnego

Opis algorytmu:

Szyfr Cezara polega na zamianie każdej litery tekstu jawnego na literę tekstu szyfrowanego przesuniętego w alfabecie o stałą liczbę znaków zwaną kluczem.

Przykładowo dla klucza $k = 3$, każdą literę alfabetu z tabeli 1 należy zamienić odpowiednią literą z tabeli 2. Natomiast dla klucza $k = -3$, każdą literę alfabetu z tabeli 1 należy zamienić odpowiednią literą z tabeli 3.

Tabela 1. Szyfrowane litery

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Tabela 2. Litery z tabeli 1 po zaszyfrowaniu kluczem $k = 3$

d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Tabela 3. Litery z tabeli 1 po zaszyfrowaniu kluczem $k = -3$

x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Założenia dotyczące algorytmu:

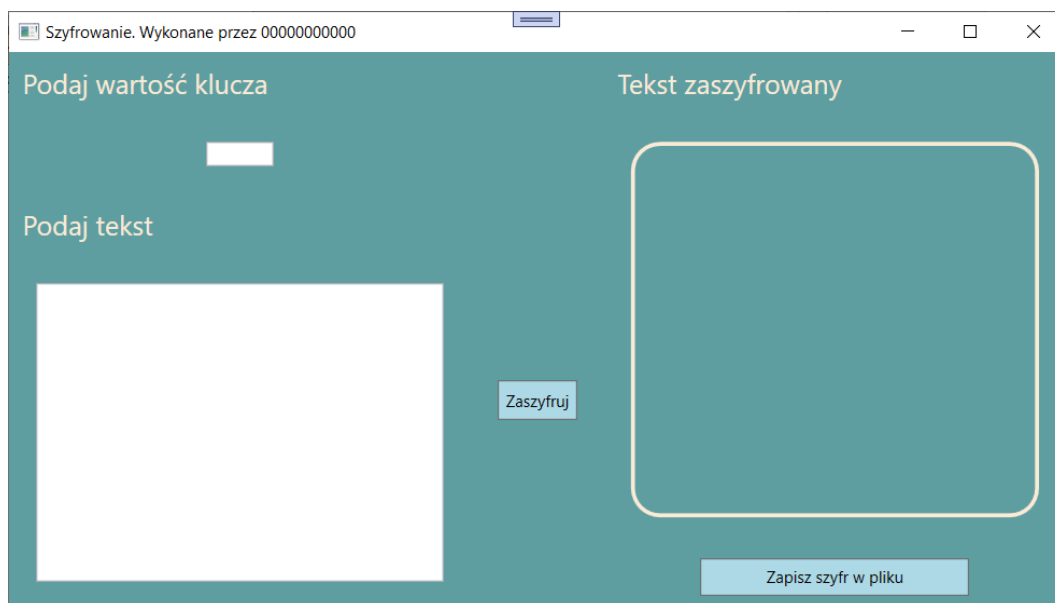
- Szyfr działa jedynie dla małych liter podstawowego alfabetu łacińskiego, przedstawionego w tabeli 1 (kody ASCII od 97 do 122)
- Znak spacji pozostaje w zaszyfrowanym tekście bez zmian
- Tekst jawny i zaszyfrowany są dowolnego typu napisowego, ich długość zależy od wczytanego tekstu
- Klucz szyfrowania jest liczbą całkowitą (wartości dodatnie i ujemne, może być większy od 26). Dla klucza $k = 0$ tekst jawny jest równy tekstowi zaszyfrowanemu
- Tekst oraz klucz są wczytywane z klawiatury w programie głównym
- W podejściu obiektowym szyfrowany tekst i klucz mogą być polami klasy
- Nie jest wymagana walidacja tekstu jawnego – zakłada się, że wprowadzono go bez błędów (tylko małe litery i spacje)

Sprawdź działanie metody lub funkcji. Program powinien nawiązać czytelną komunikację z użytkownikiem. Po uruchomieniu programu wykonaj zrzuty ekranu dokumentujące interakcje programu dla różnych wartości klucza (0, ujemna, dodatnia). Zrzuty ekranu powinny obejmować cały obszar ekranu, z widocznym paskiem zadań oraz środowiskiem programistycznym. Liczba zrzutów ekranu powinna odpowiadać wszystkim możliwym interakcjom użytkownika z programem. Zrzuty ekranu zapisz w folderze *konsolowa* pod nazwami *konsola1.png*, *konsola2.png*, itd.

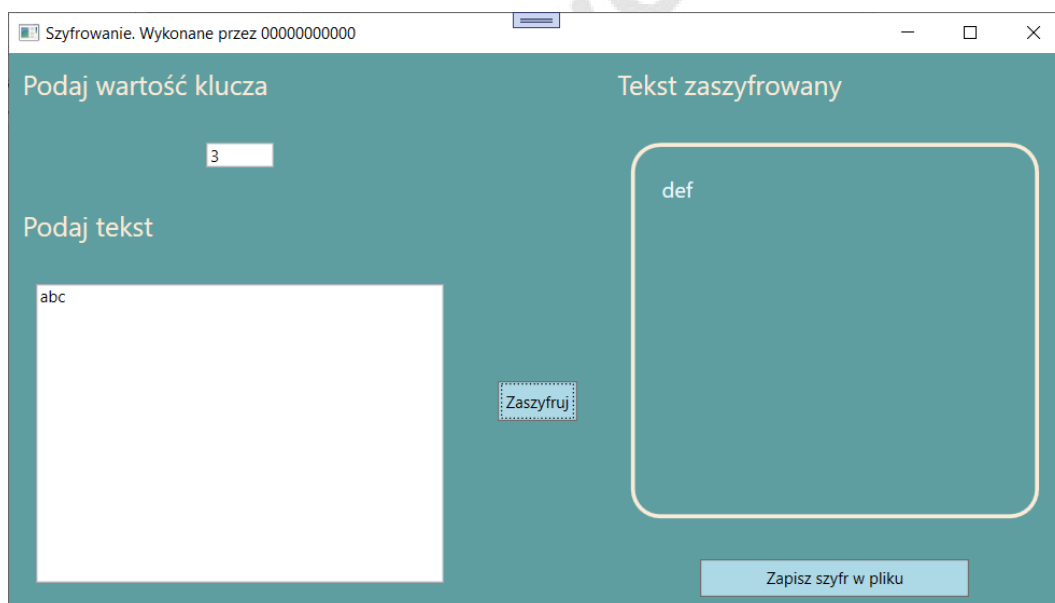
Kod aplikacji przygotuj do nagrania na płytę. W folderze *konsolowa* powinno znaleźć się archiwum całego projektu o nazwie *konsola.zip*, skopiowany z projektu plik z kodem źródłowym programu oraz plik wykonywalny, jeżeli istnieje.

Część II. Aplikacja desktopowa

Za pomocą dostępnego na stanowisku egzaminacyjnym środowiska programistycznego wykonaj aplikację desktopową będącą interfejsem graficznym programu utworzonego w konsoli. Na obrazie 2 przedstawiono ideę aplikacji desktopowej. W zależności od użytego środowiska programistycznego wygląd może nieznacznie się różnić. Do wykonania zadania można posłużyć się funkcją / metodą z aplikacji konsolowej.



Obraz 1. Stan początkowy



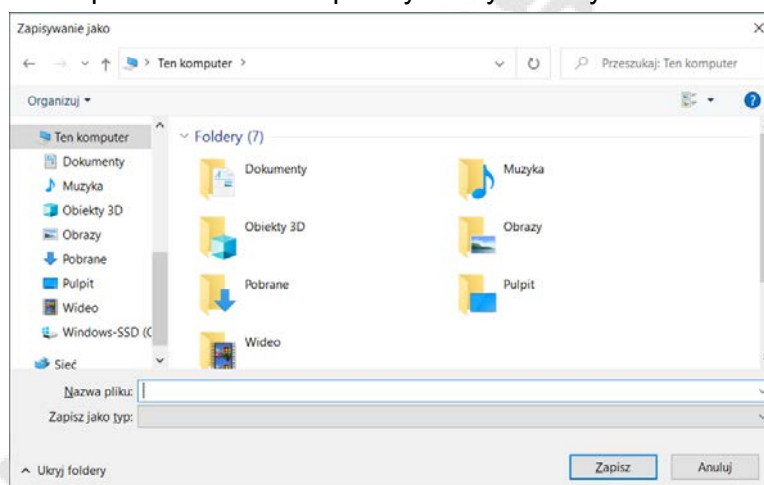
Obraz 2. Działanie aplikacji

Opis wyglądu aplikacji:

- Tytuł okna „Szyfrowanie. Wykonane przez ”, dalej wstawiony numer zdającego
- Kolor tła okna CadetBlue (#5F9EA0)
- Kontrolki rozmieszczone zgodnie z obrazem 1, z określonym formatowaniem:
 - Napisy: „Podaj wartość klucza”, „Podaj tekst”, „Tekst zaszyfrowany” w kolorze czcionki AntiqueWhite (#FAEBD7)
 - Pole edycyjne do wprowadzania wartości klucza
 - Pole edycyjne wielowierszowe do wprowadzania tekstu z włączonym zawijaniem wierszy
 - Pole, w którym wyświetlony jest zaszyfrowany tekst o cechach:
 - Kolor czcionki AliceBlue (#F0F8FF)
 - Obramowanie: linia ciągła koloru AntiqueWhite (#FAEBD7), rogi zaokrąglone
- Przyciski o kolorze tła LightBlue (#ADD8E6)

Założenia dotyczące działania aplikacji:

- Klucz szyfrowania i tekst szyfrowany są pobierane z odpowiednich kontrolek
- Szyfrowanie odbywa się po wprowadzeniu tekstu i klucza i wybraniu przycisku „Zaszyfruj”
- Jeżeli w polu przeznaczonym na wpisanie klucza nie została wpisana wartość liczbowa całkowita, po naciśnięciu przycisku, do szyfrowania, należy zastosować wartość klucza 0
- Po wybraniu przycisku „Zapisz szyfr w pliku”, zostanie otwarte systemowe okno dialogowe (obraz 3) do zapisu pliku. W pliku ma zostać zapisany zaszyfrowany tekst.



Obraz 3. Zapisywanie szyfru do pliku

Aplikacja powinna być zapisana czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

Podjmij próbę kompilacji i uruchomienia aplikacji. Wykonaj zrzuty ekranu dokumentujące wszystkie stany aplikacji. Zrzuty ekranu powinny obejmować cały obszar ekranu, z widocznym paskiem zadań oraz środowiskiem programistycznym. Zrzuty zapisz w folderze *desktopowa* pod nazwami *desktop1.png*, *desktop2.png*, itd.

Kod aplikacji przygotuj do nagrania na płytę. W folderze *desktopowa* powinno znaleźć się archiwum całego projektu o nazwie *desktopowa.zip*, skopiowane z projektu pliki źródłowe, które były modyfikowane w czasie egzaminu (interfejs użytkownika i logika aplikacji) oraz plik wykonywalny, jeżeli istnieje.

Część III. Testy jednostkowe

Wykonaj testy jednostkowe funkcji lub metody realizującej szyfr Cezara z aplikacji konsolowej. Do napisania testów wykorzystaj dostępną na stanowisku egzaminacyjnym bibliotekę do testów jednostkowych dla języka programowania, w którym została napisana aplikacja konsolowa. W testach jednostkowych należy sprawdzić przypadki testowe opisane w tabeli 4.

Tabela 4. Scenariusz testowania z przypadkami testowymi

Nr testu	Cel testu (Jaka sytuacja będzie testowana)	Dane wejściowe		Oczekiwana wartość po zaszyfrowaniu
		Tekst jawny	Klucz	
1	Dane podstawowe (klucz dodatni, wartość mniejsza od długości alfabetu)	abc	3	def
2	„Zawijanie” (gdy litery w tekście i klucz wychodzą poza alfabet)	xyz	3	abc
3	Odszyfrowanie (klucz ujemny)	def	-3	abc
4	Klucz większy niż długość alfabetu	abc	29	def
5	Spacja w tekście	ab cd	2	cd ef

Nazwy metod testujących należy dobrać tak, aby wyrażały cel danego testu. Nazwy metod należy zapisać zgodnie z konwencją nazewnictwa w danym języku programowania.

Każdy przypadek testowy powinien być sprawdzany za pomocą asercji oddzielną metodą. Metody testujące należy zapisać w oddzielnym pliku w lokalizacji zgodnej z wymaganiami danego języka programowania i wykorzystanej biblioteki do zapisu testów jednostkowych.

Po napisaniu metod testujących, uruchom wszystkie metody. Wykonaj zrzut/zrzuty ekranu dokumentujące uruchomienie testów, zrzuty zapisz w folderze *testy* pod nazwami *test1.png*, *test2.png*, itd. Na zrzutach powinny być widoczne wyniki działania metod testujących. Zrzuty ekranu powinny obejmować cały obszar ekranu, z widocznym paskiem zadań oraz środowiskiem programistycznym.

Jeżeli metoda szyfrująca została napisana w części I niepoprawnie, testy powinny wykazać błędy. W tej części egzaminu ocenie podlegać będzie zapis i uruchomienie testu, a nie metoda szyfrująca.

W folderze *testy* zapisz bezpośrednio pliki, których treść była modyfikowana oraz zrzuty ekranu. Utwórz archiwum ZIP zawierające cały projekt i zapisz je w folderze *testy*.

Utwórz plik z dokumentacją i nazwij go *egzamin*. Dokument powinien zawierać informacje o narzędziach wykorzystanych na egzaminie:

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

Dokument *egzamin* zapisz w folderze z numerem zdającego.

UWAGA: Nagraj płytę z rezultatami pracy. W folderze z numerem zdającego powinny znajdować się podfoldery desktopowa, konsolowa, testy oraz plik egzamin. W folderze desktopowa: spakowany cały projekt aplikacji desktopowej, pliki ze źródłami interfejsu i logiki, opcjonalnie plik wykonywalny, zrzuty ekranu. W folderze konsolowa: spakowany cały projekt aplikacji konsolowej, pliki źródłowe, opcjonalnie plik wykonywalny, zrzuty ekranu. W folderze testy: spakowany projekt z testami, pliki z zapisem metod testujących, zrzuty ekranu. Opisz płytę numerem zdającego i pozostaw na stanowisku, zapakowaną w pudełku wraz z arkuszem egzaminacyjnym.

Czas przeznaczony na wykonanie zadania wynosi 180 minut.

Ocenie podlegać będą 4 rezultaty:

- implementacja, kompilacja, uruchomienie programów,
- aplikacja konsolowa,
- aplikacja desktopowa,
- testy aplikacji.

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

[illegible]

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

.....
Czytelny podpis Przewodniczącego ZN