

Wydział Budowy Maszyn i Informatyki

**CYBERBEZPIECZEŃSTWO**  
**(ćwiczenia projektowe №1)**

**Temat ćwiczenia: Systemy bezpieczeństwa oparte na hasłach**

## 1. Wprowadzenie

Hasła to powszechnie wykorzystywany, prosty i wszystkim znany sposób chronienia dostępu do poufnych danych i usług. Aby jednak działały skutecznie, muszą być tworzone i stosowane w sposób prawidłowy, w tym dostosowany do aktualnych wymogów technicznych i organizacyjnych.

Bezpieczny dostęp do informacji przeznaczonych dla określonej grupy osób zawsze wymaga weryfikacji, czy osoba ubiegająca się o ten dostęp jest tym, za kogo się podaje. Proces udowadniania swojej tożsamości nazywamy uwierzytelnianiem. Najczęściej wymaga on przekazania i weryfikacji tzw. sekretu. Może nim być np. hasło, czy kod PIN. Najważniejszą cechą sekretu jest jego tajność – powinien on być znany tylko uprawnionym osobom. Jest to prosty technicznie mechanizm, który ma jednak pewne mankamenty. Szybki rozwój technologii oraz przenoszenie coraz większej części naszego życia do sfery cyfrowej spowodowały, że liczba systemów, w których musimy się regularnie uwierzytelniać, znacząco wzrosła.

## 2. Zadanie do zrealizowania:

Napisz program, który implementuje następujące zasady dla systemu bezpieczeństwa:

1. Program powinien zapewniać pracę w dwóch rolach: administratora (użytkownik o stałej nazwie ADMIN) oraz zwykłego użytkownika.
2. W rolę administratora program musi zawierać następujące funkcje:
  - zmienianie hasła dla konta administratora (jeśli stare hasło zostało wprowadzone poprawnie);
  - modyfikować szczegóły konta, np. pełną nazwę użytkownika czy hasło;
  - dodawać nowych użytkowników;
  - przeglądać listę użytkowników;
  - blokować konta użytkowników oraz blokować ograniczenie wybranych haseł;
  - usuwać konta użytkowników;
  - włączyć / wyłączyć ograniczenia haseł wybranych przez użytkownika (zgodnie z zadaniem indywidualnym). *Rys.1 – przykład prototypowania.*;
  - ustawić ważność hasła użytkownika. Po wygaśnięciu hasła (np. po upływie dni ustalonych), użytkownik podaje nowe hasło, które musi różnić się od wszystkich poprzednich. *Rys.1 – przykład prototypowania.*
  - zakończenie pracy z programem.
3. W roli użytkownika program powinien zawierać tylko funkcje zmiany hasła użytkownika (jeśli stare hasło jest wpisane poprawnie) i zakończenia pracy.
4. Po uruchomieniu program powinien poprosić użytkownika podać swój identyfikator i hasło do konta w specjalnym oknie logowania. Wprowadzając hasło, jego znaki należy zawsze zastąpić wyświetlanym na ekranie symbolem ”\*”.
5. Komunikat w przypadku wprowadzenia niepoprawnego identyfikatora lub hasła: „Login lub Hasło niepoprawny”.
6. Przy pierwszym logowaniu system powinien prosić o zmianę hasła dostępu ustalonego przez administratora na hasło własne, znane tylko użytkownikowi, utworzone według zadania indywidualnego. Nowe hasło należy podać oraz powtórzyć.
7. Należy stosować bezpieczny algorytm hashujący do przechowywania haseł.


## 3. Zadania indywidualne (Ograniczenia dotyczące haseł użytkowników)


Hasło musi zawierać

1. co najmniej 8 znaków, co najmniej jedną wielką literę, co najmniej jeden znak specjalny;
2. co najmniej 12 znaków, co najmniej jedną cyfrę, co najmniej jeden znak specjalny;
3. brak powtarzających się znaków;
4. co najmniej jedną wielką literę i 2 cyfr;
5. co najmniej jedną małą literę, co najmniej jeden znak specjalny;
6. co najmniej 14 znaków, co najmniej jedną cyfrę;
7. co najmniej jedną wielką literę, jedną małą literę i 3 cyfr;
8. znaki z kategorii: cyfry, znaki specjalne
9. znaki z kategorii: wielkie litery, małe litery, znaki specjalne



10. znaki z kategorii: małe litery, cyfry

## 4. Prototypowanie

**Ustawienia** 




 **Bezpieczeństwo**

Minimalna długość i złożoność hasła dla użytkownika

Długość hasła:   

Złożoność hasła: ☐ Znaki specjalne lub cyfry ☐ Małe i wielkie litery

Ważność hasła użytkownika

  dni 

Administracja

☐ Zablokuj ustawienia użytkownika na tym komputerze

Rys.1