



# Podstawy programowania III

## WYKŁAD 5

Jan Kazimirski



# Projekt: „Katalog książek elektronicznych”



## Założenia projektu

- Aplikacja będzie służyła do zarządzania zbiorem książek w postaci elektronicznej.
- Aplikacja będzie przechowywała informacje o książkach oraz kategoriach do których książka została przypisana.
- Aplikacja będzie umożliwiała zarządzanie książkami (CRUD), kategoriami (CRUD), przypisywaniem książek do kategorii, oraz wyszukiwaniem książek według zadanych kryteriów



## Założenia projektu c.d.

- W celu zabezpieczenia przed przypadkowym usunięciem lub edycją danych, aplikacja będzie działała w dwóch trybach: dodawanie/wyszukiwanie (domyślnie) lub edycja. Edytowanie i usuwanie rekordów będzie dostępne tylko w trybie edycji.
- Przechowywane informacje o książce: ID, autor, tytuł, wydawca, rok wydania, słowa kluczowe, ścieżka do pliku, typ pliku.
- Każdą książkę będzie można przypisać do jednej lub wielu kategorii (relacja wiele-do-wielu).



## Model obiektowy vs. model relacyjny

- Stosowany obecnie obiektowy paradygmat programowania nie ma zastosowania do baz danych.
- Obiektowe bazy danych istnieją, ale nie cieszą się popularnością.
- Stosowane obecnie bazy danych są bazami relacyjnymi.



# Model obiektowy vs. model relacyjny c.d.

- Tworzenie aplikacji wykorzystującej bazę danych wymaga zastosowania obu paradygmatów i „punktów łączenia”

MODEL  
RELACYJNY



MODEL  
OBIEKTOWY

## Diagram ERD

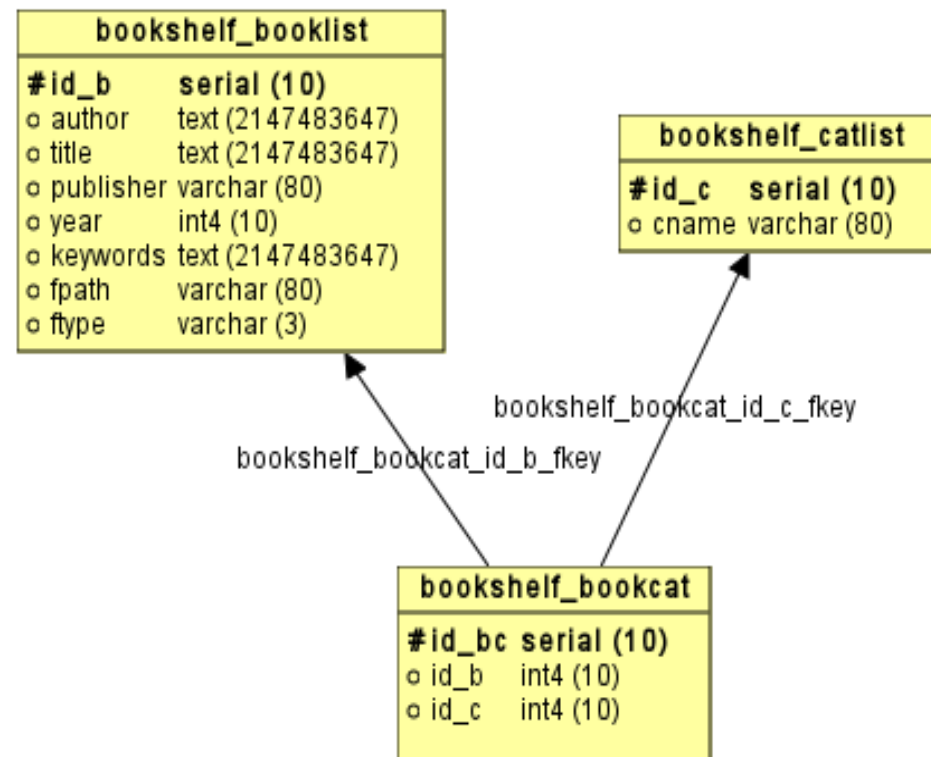
Model danych aplikacji

Diagram ERD (Entity-Relationship Diagram)

Tabela **bookshelf\_booklist**  
- informacje o książkach

Tabela **bookshelf\_catlist**  
- lista kategorii

Tabela **bookshelf\_bookcat**  
- Realizacja powiązania wiele-do-wielu – przypisywanie książek do kategorii.





# Tworzenie tabel w bazie danych

```
create table bookshelf_booklist (  
  id_b serial primary key,  
  author text,  
  title text,  
  publisher varchar(80),  
  year int,  
  keywords text,  
  fpath varchar(80),  
  ftype varchar(3)  
);
```





# Tworzenie tabel w bazie danych c.d.

```
create table bookshelf_catlist (  
  id_c serial primary key,  
  cname varchar(80) unique not null  
);  
  
create table bookshelf_bookcat (  
  id_bc serial primary key,  
  id_b int references bookshelf_booklist(id_b),  
  id_c int references bookshelf_catlist(id_c)  
);
```



## Aktorzy i przypadki użycia

- Aktorzy: Użytkownik
- Przypadki użycia:
  - Zarządzaj książkami (CRUD)
  - Zarządzaj kategoriami (CRUD)
  - Szukaj książki (ogólny)



## Przypadki użycia c.d.

- Przypadek CRUD – Zarządzaj <obiektem>:
  - Dodaj <obiekt>
  - Wyświetl <obiekt>
  - Edytuj <obiekt>
  - Usuń <obiekt>
- Przypadek „Szukaj książki”:
  - Wyszukiwanie po numerze ID, autorze, tytule, słowach kluczowych (częściowe dopasowanie)
  - Możliwość ograniczenia wyszukiwania do określonych kategorii.

# Diagram przypadków użycia





## Scenariusz 1

**Nazwa scenariusza:** Włącz tryb edycji

**Warunki wstępne:** Wyłączony tryb edycji, wyświetlony przycisk włączania trybu edycji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk włączania trybu edycji
2. System włącza tryb edycji i wyświetla przycisk wyłączenia trybu edycji

**Warunki końcowe:** Tryb edycji włączony, wyświetlony przycisk wyłączenia edycji.



## Scenariusz 2

**Nazwa scenariusza:** Wyłącz tryb edycji

**Warunki wstępne:** Włączony tryb edycji, wyświetlony przycisk wyłączenia edycji.

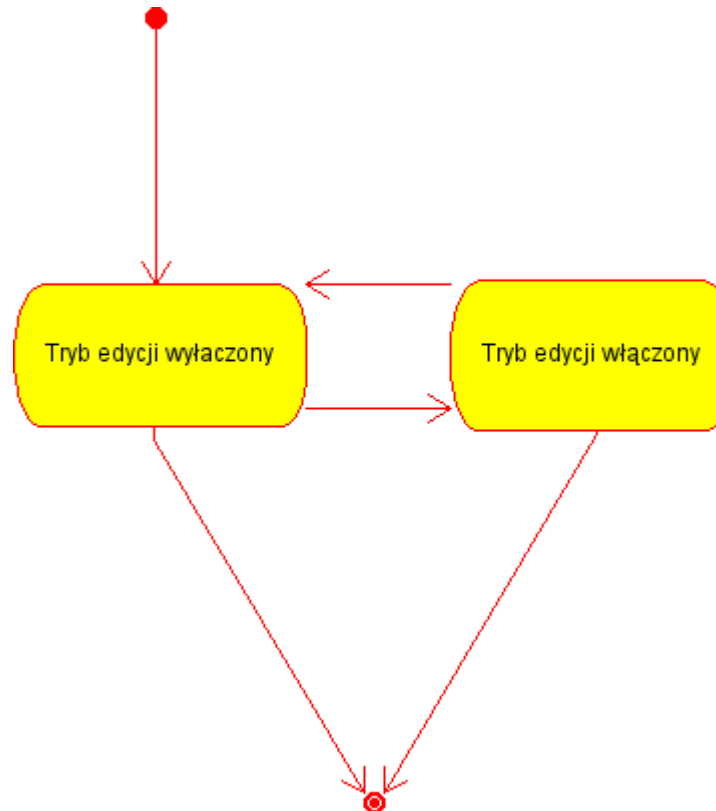
**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk wyłączenia trybu edycji
2. System wyłącza tryb edycji i wyświetla przycisk włączenia trybu edycji

**Warunki końcowe:** Tryb edycji wyłączony, wyświetlony przycisk włączenia edycji.



# Tryby pracy aplikacji – diagram stanów





## Scenariusz 3

**Nazwa scenariusza:** Dodaj książkę

**Warunki wstępne:** Wyświetlony główny panel aplikacji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk dodania książki.
2. System wyświetla formularz wprowadzania danych książki i listę kategorii.
3. Użytkownik wprowadza dane książki i zaznacza kategorie do których książka należy. Użytkownik zatwierdza dane.

**Warunki końcowe:** Nowa książka wprowadzona do systemu.





## Scenariusz 4

**Nazwa scenariusza:** Dodaj kategorię

**Warunki wstępne:** Wyświetlony główny panel aplikacji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk dodania kategorii
2. System wyświetla formularz dodawania nowej kategorii
3. Użytkownik wprowadza nazwę kategorii i zatwierdza dane

**Warunki końcowe:** Nowa kategoria wprowadzona do systemu.



## Scenariusz 5

**Nazwa scenariusza:** Wyświetl listę kategorii

**Warunki wstępne:** Wyświetlony główny panel aplikacji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk wyświetlenia listy kategorii
2. System wyświetla listę zdefiniowanych kategorii

**Warunki końcowe:** Wyświetlona lista kategorii

**Rozszerzenia:** Usuń kategorię, edytuj kategorię



## Scenariusz 6

**Nazwa scenariusza:** Usuń kategorię

**Warunki wstępne:** Wyświetlona lista zdefiniowanych kategorii, włączony tryb edycji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk usunięcia określonej kategorii
2. System usuwa podaną kategorię
3. System ponownie wyświetla listę kategorii

**Warunki końcowe:** Wybrana kategoria zostaje usunięta.



## Scenariusz 7

**Nazwa scenariusza:** Edytuj kategorię

**Warunki wstępne:** Wyświetlona lista zdefiniowanych kategorii, włączony tryb edycji

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk edycji wybranej kategorii
2. System wyświetla formularz edycji wybranej kategorii
3. Użytkownik zmienia nazwę kategorii i zatwierdza zmianę
4. System ponownie wyświetla listę kategorii.

**Warunki końcowe:** Nazwa wybranej kategorii została zmieniona.



## Scenariusz 8

**Nazwa scenariusza:** Szukaj książek

**Warunki wstępne:** Wyświetlony główny panel aplikacji.

**Główny przebieg zdarzeń:**

1. Użytkownik wybiera przycisk szukania książki
2. System wyświetla formularz określający warunki wyszukiwania
3. Użytkownik wprowadza warunki wyszukiwania
4. System wyświetla listę książek spełniających warunki wyszukiwania

**Warunki końcowe:** Wyświetlona lista książek spełniająca warunki wyszukiwania.

**Rozszerzenia:** Usuń książkę, edytuj książkę



## Scenariusz 8 c.d.

### Notatki:

#### 1. Przypadek ogólny. Przypadki szczegółowe

- szukaj w kategoriach
- szukaj po autorze
- szukaj po tytule
- szukaj po słowach kluczowych

pozwalają określić kryteria szukania

#### 2. Przypadek użycia rozpoczynany automatycznie w momencie startu aplikacji.



## Scenariusz 9

**Nazwa scenariusza:** Usunąć książkę

**Warunki wstępne:** Wyświetlona lista książek, włączony tryb edycji.

**Główny przebieg zdarzeń:**

1. Użytkownik wybiera przycisk usunięcia wybranej książki
2. System usuwa wybraną książkę i ponownie wyświetla listę książek

**Warunki końcowe:** Wybrana książka usunięta z systemu



## Scenariusz 10

**Nazwa scenariusza:** Edytuj książkę

**Warunki wstępne:** Wyświetlona lista książek, włączony tryb edycji.

**Główny przebieg zdarzeń:**

1. Użytkownik naciska przycisk edycji wybranej książki
2. System wyświetla formularz edycji wybranej książki
3. Użytkownik wprowadza i zatwierdza zmiany
4. System ponownie wyświetla listę książek

**Warunki końcowe:** Dane wybranej książki zostały zmodyfikowane.





# Dygresja – przypadki użycia i scenariusze

*Źródło: Alistair Cockburn, „Inżynieria oprogramowania. Jak pisać efektywne przypadki użycia”, WNT, 2004*

1. Przypadek użycia to pisane prozą opowiadanie.
2. Przypadki użycia powinny być czytelne.
3. Czynności powinny być wyrażone w sposób czytelny (kto robi co?)
4. Długie przypadki użycia powinny być upraszczane (pow. 9 kroków scenariusza głównego).



## Dygresja – przypadki użycia i scenariusze c.d.

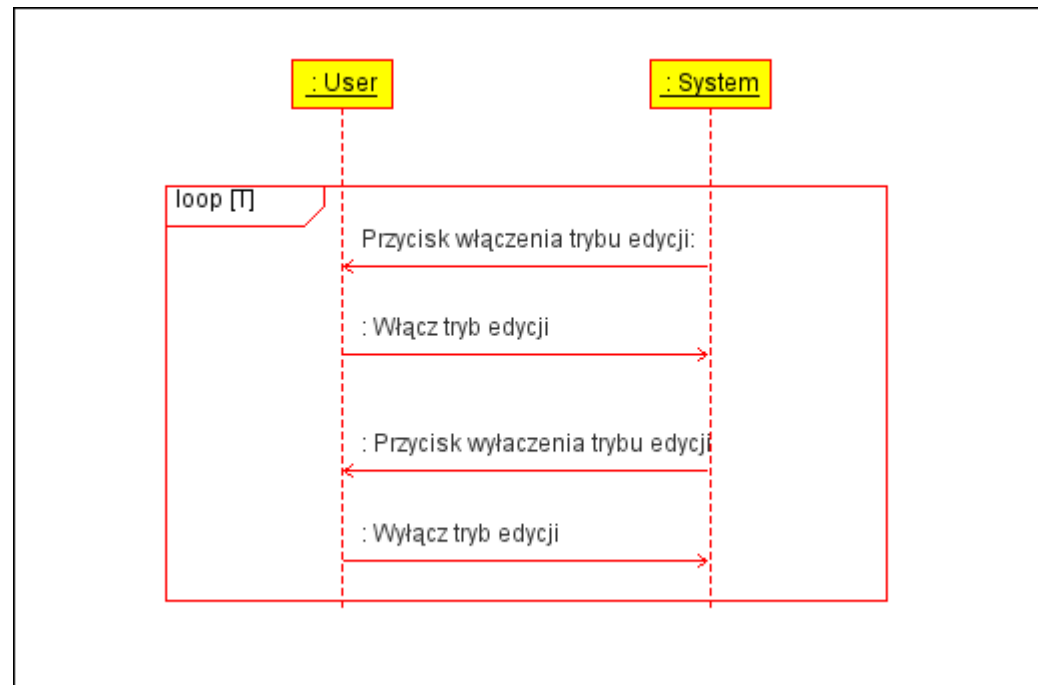
5. Przypadek użycia powinien jasno określać warunki początkowe.
6. Przypadek użycia nie powinien definiować szczegółów interfejsu graficznego.
7. Liczne konstrukcje „Jeżeli ... to ...” w scenariuszu należy zastępować rozszerzeniami.
8. Nie należy zastępować scenariuszy diagramami.



## Dygresja – przypadki użycia i scenariusze c.d.

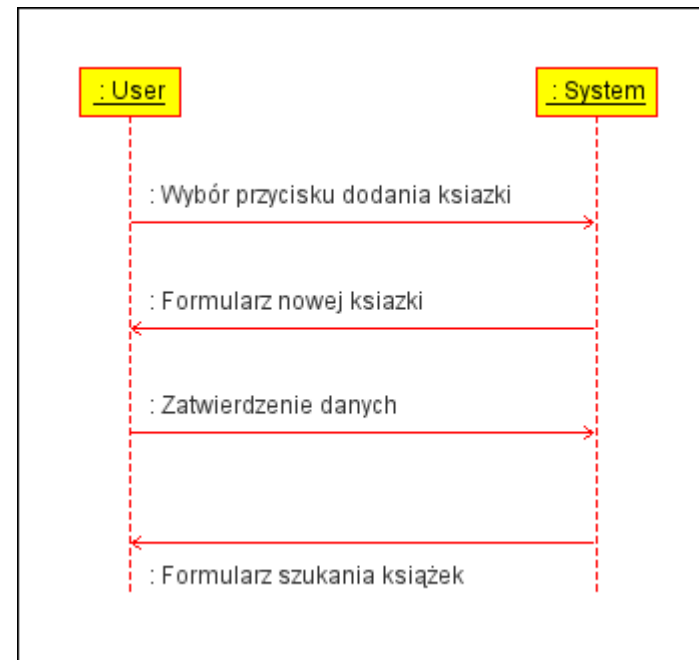
9. Scenariusze powinny uwzględniać sytuacje awaryjne (co może pójść źle?).
10. Lepiej napisać trochę za mało niż zbyt dużo.
11. Diagram przypadków użycia to tylko „widok z lotu ptaka”. Ważne są scenariusze.
12. Uwaga na narzędzia. Często najlepszy jest papier i ołówek.

# Diagram sekwencji – scenariusze 1,2



Zmiana trybu pracy: tryb edycji  
wyłączony lub włączony

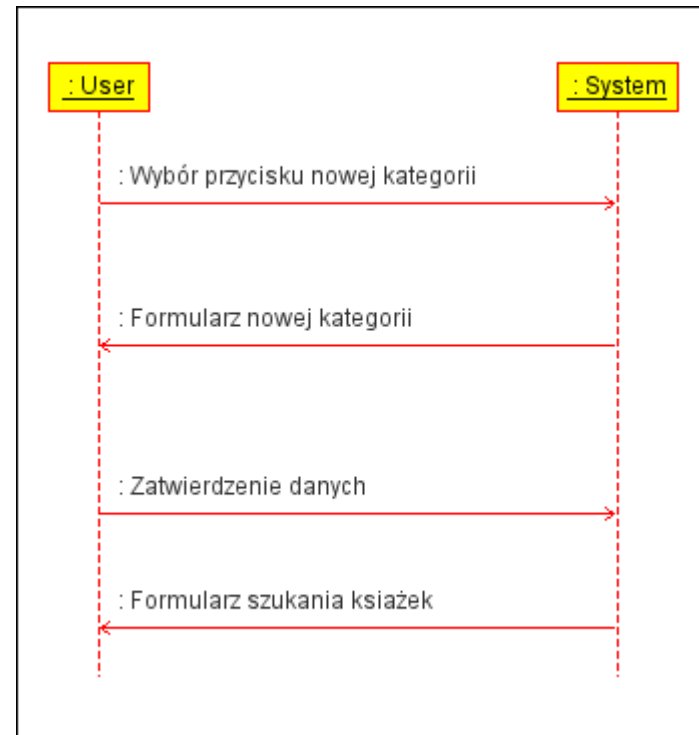
## Diagram sekwencji – scenariusz 3



Dodanie książki



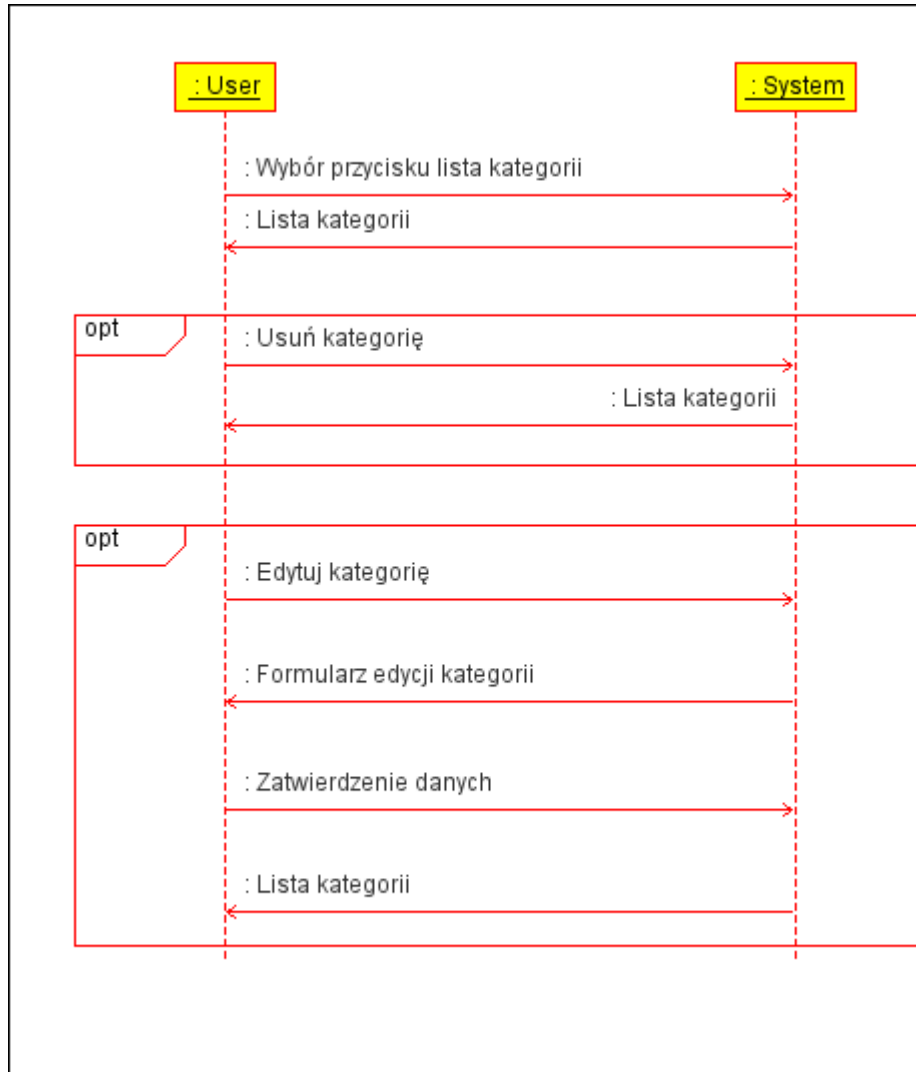
## Diagram sekwencji – scenariusz 4



Dodanie kategorii



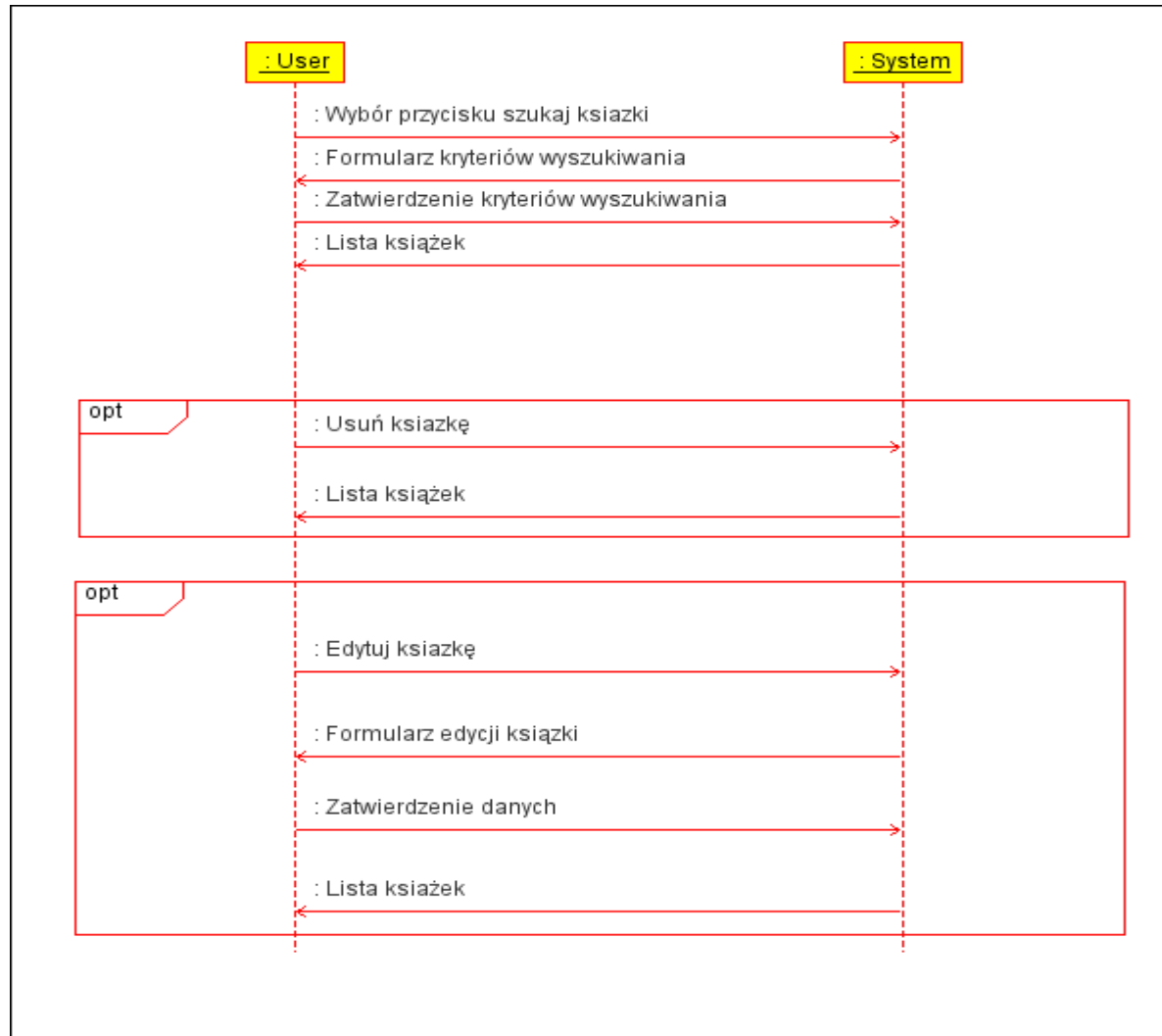
## Diagram sekwencji – scenariusze 5-7



Lista kategorii,  
dodaj, usuń  
kategorię



## Diagram sekwencji – scenariusze 8-10



Lista książek zgodna z kryteriami szukania, usuń, edytuj książkę.





## Identyfikacja klas, atrybutów i metod

- Metoda „Rzeczowników i czasowników”
  - Z poszczególnych scenariuszy wynotowujemy rzeczowniki i związane z nimi czasowniki
  - Usuwamy powtórzenia
  - Ustalamy potencjalnych kandydatów na klasy (rzeczowniki) i metody (czasowniki).



## Identyfikacja ... c.d.

Książka

Kategoria

Lista książek

Lista kategorii

Tryb edycji

Panel główny

Kryteria wyszukiwania

F. nowej książki

F. nowej kategorii

F. edycji kategorii

F. edycji książki

F. kryteriów szukania.

P. włączenia trybu edycji

P. wyłączenia trybu edycji

P. szukania książki

P. dodania książki

P. dodania kategorii

P. wyświetlenia listy kategorii

P. usunięcia kategorii

P. edycji kategorii

P. usunięcia książki

P. edycji książki



## Identyfikacja ... c.d.

Czynności:

**wszystkie przyciski** – wyświetlanie, wybór

**wszystkie formularze** – wyświetlanie, odbiór (zatwierdzenie)

**panel główny** – wyświetlanie

**tryb edycji** – włączanie wyłączenie

**książka/lista książek** – wyświetlanie, usuwanie, dodawanie, aktualizacja,  
szukanie wg. kryteriów

**kategoria/lista kategorii** - wyświetlanie, usuwanie, dodawanie, aktualizacja

**kryteria wyszukiwania**



# Struktura logiczna i fizyczna aplikacji

- Struktura logiczna systemu – podział systemu na klasy i opis ich wzajemnych interakcji – architektura systemu.
- Struktura fizyczna – podział systemu na fizyczne jednostki – pliki, komponenty, biblioteki, oraz ich wzajemne relacje – sposób realizacji systemu.
- Komponent – najmniejszy element struktury fizycznej – może zawierać klasy, funkcje, wyliczenia itp.



# Dygresja – Hierarchia fizyczna projektu

*Źródło: John Lakos, „C++. Projektowanie systemów informatycznych. Vademecum profesjonalisty”, Helion, 2004*

Hierarchia fizyczna projektu – zbiór relacji „zależy od” między komponentami.

Złożoność i struktura relacji między komponentami – wpływ na czas kompilacji/wykonania, a przede wszystkim na możliwości testowania („testowalność”).



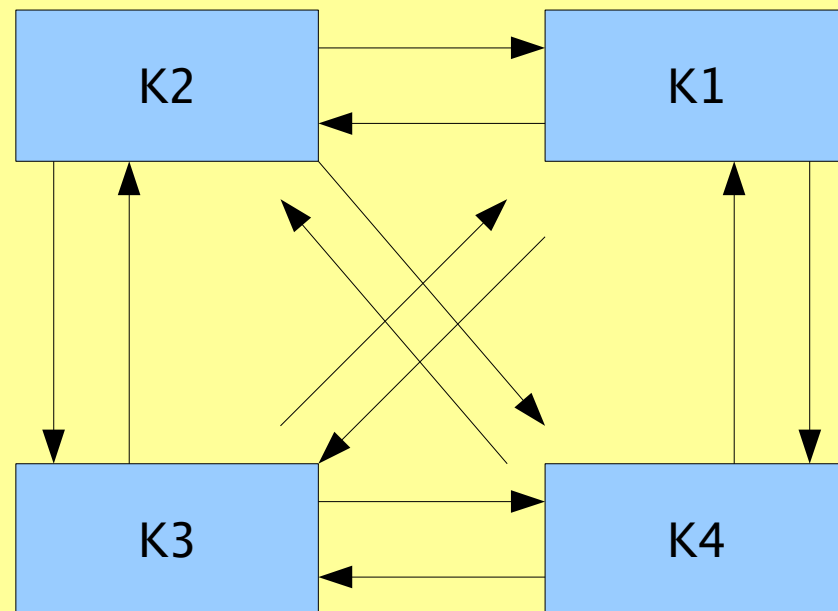
## Dygresja – Hierarchia fizyczna projektu c.d.

Analogia „samochodowa”

- Klient kupuje samochód – interesuje go działanie auta jako całości, nie musi (i nie chce znać szczegółów)
- Mechanik – testuje poszczególne elementy, w miarę możliwości „w oderwaniu” od całości.
- Samochód jako system – skomplikowany system złożony z wielu komponentów.
- Komponenty powinny być w miarę możliwości niezależne (możliwość uruchomienia/testowania pod nieobecność innych komponentów)

# Dygresja – Hierarchia fizyczna projektu c.d.

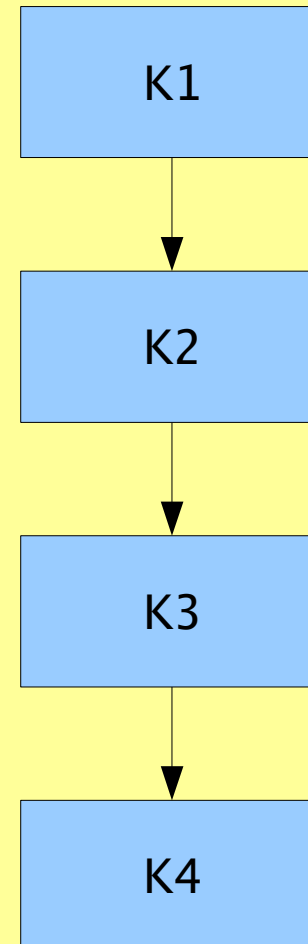
Projekt ze ściśle powiązаныmi komponentami, w tym zależności cykliczne.  
Długi czas kompilacji/wykonania  
Trudne testowanie



# Dygresja – Hierarchia fizyczna projektu c.d.

Mało zależności między komponentami ale sztywna struktura.

Wymagana kolejność testowania: K4, K3, K2, K1







## Dygresja – Hierarchia fizyczna projektu c.d.

Brak zależności między komponentami.  
Szybka kompilacja/wykonanie.  
Dowolna kolejność testowania.

K1

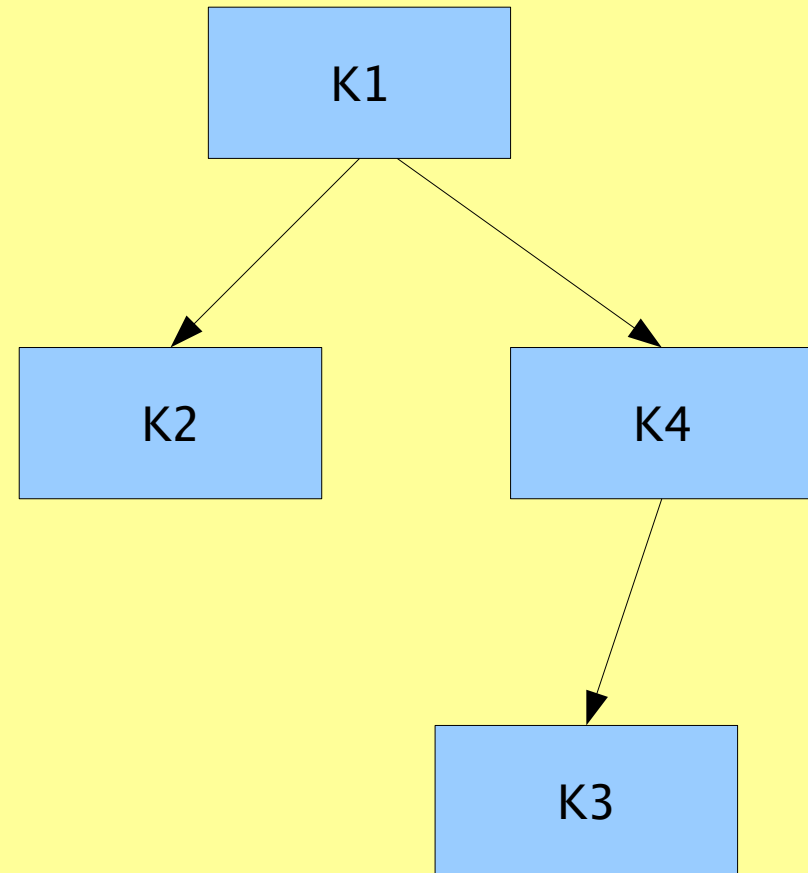
K2

K3

K4

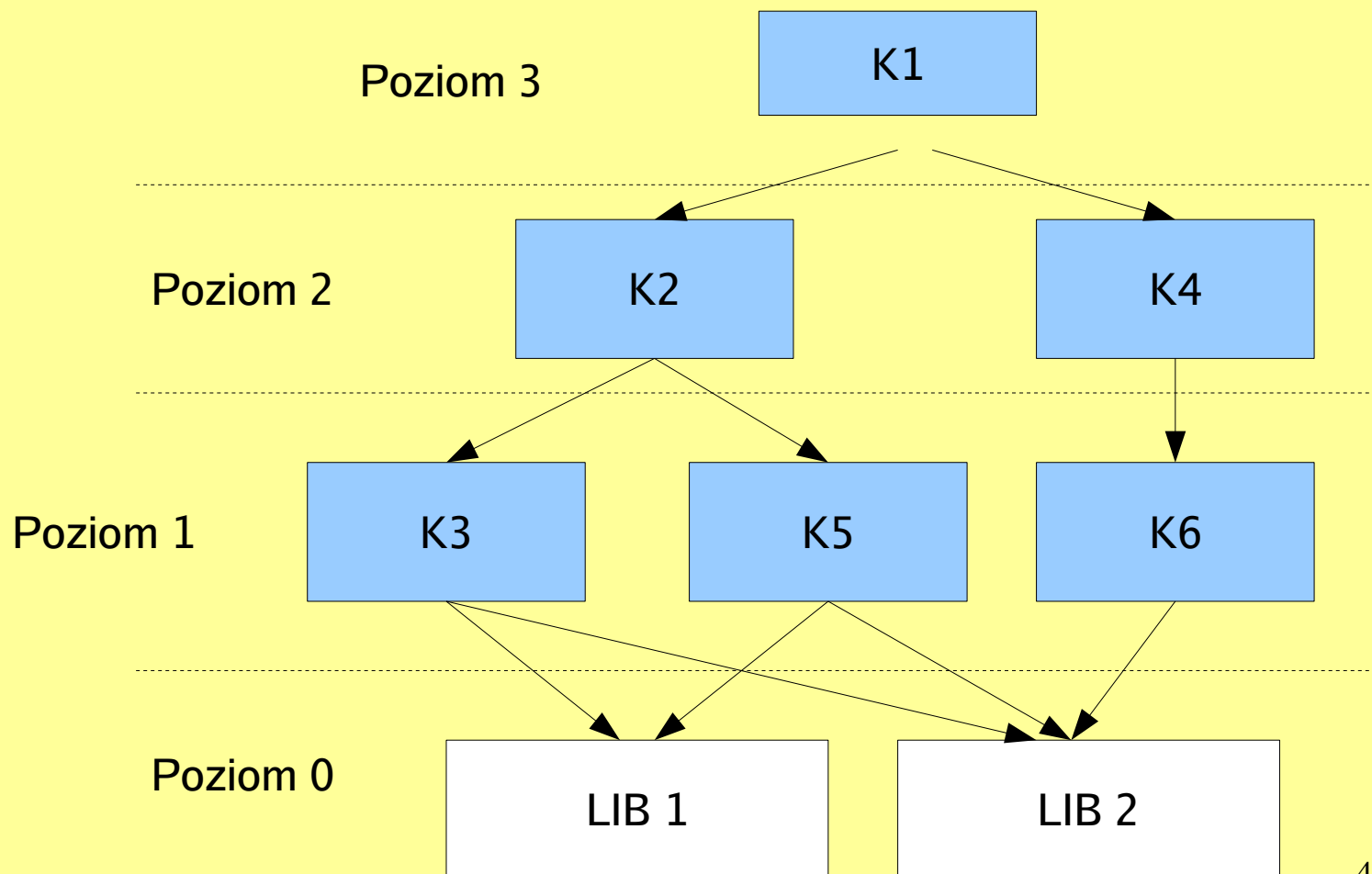
# Dygresja – Hierarchia fizyczna projektu c.d.

Typowy „dobry” projekt –  
Struktura drzewiasta.  
Brak zależności cyklicznych  
Niewiele zależności.  
Pewna swoboda testowania z  
możliwością częściowego  
zrównoleglenia testów



# Dygresja – Hierarchia fizyczna projektu c.d.

Przykład podziału struktury fizycznej na poziomy.



# PRACA DOMOWA

