Proof-of-Concept dla

[nazwa technologi/narzędzia]

Opracował: [Imię] [Nazwisko] [Numer Indeksu]

Spis treści

[1. Technologia 3](#_Toc2808606)

[2. Funkcjonalność języka 3](#_Toc2808607)

[Natywne wsparcie dla operacji na danych 3](#_Toc2808608)

[Biblioteka [nazwa] 4](#_Toc2808609)

[Biblioteka [nazwa] 4](#_Toc2808610)

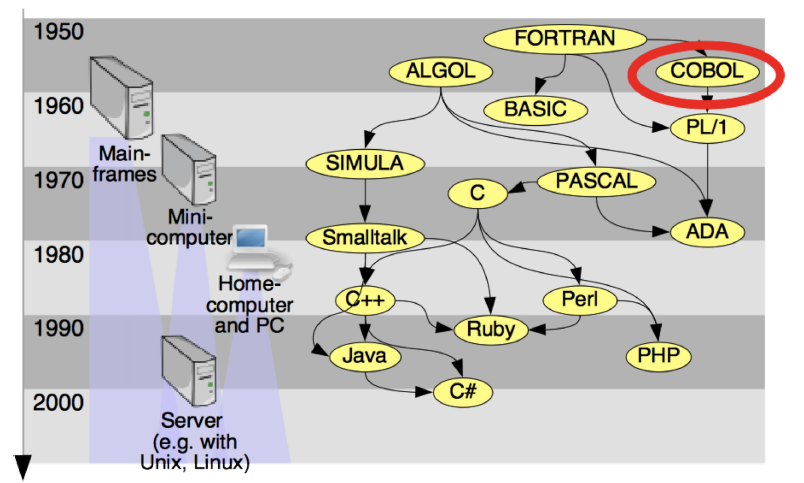
[3. Analiza SWOT 4](#_Toc2808611)

[4. Rekomendacja 4](#_Toc2808612)

# 1. Technologia

COBOL (ang. common business-oriented language) – wysokopoziomowy język programowania stworzony i używany do tworzenia aplikacji biznesowych. COBOL jest językiem imperatywnym, proceduralnym, oraz od 2002 roku, obiektowym.

W ostatnich latach zauważalny jest stopniowy spadek znaczenia COBOLa w tworzeniu nowych aplikacji. Większość wykonywanych prac programistycznych związania jest z utrzymaniem i rozwojem aplikacji istniejących. Z uwagi na jego malejącą popularność oraz mniejszą podaż programistów COBOL, część programów migrowana jest na nowe platformy, przepisywana na nowsze języki bądź zastępowana nowymi pakietami oprogramowania. Niemniej COBOL nadal odgrywa znaczącą rolę w obszarze aplikacji biznesowych na świecie.

**[[1]](#footnote-1)

# 2. Funkcjonalność języka

W tej sekcji znajduje się opis funkcjonalności dla wybranej technologii.

## Natywne wsparcie dla operacji na danych

COBOL wspiera operowanie na danych w modelu CODASYL. Operacje wstawiania danych można zrealizować za pomocą wyrażeń:

|  |
| --- |
| Kod |

Dzięki poniższemu kodu zostanie wykonana operacja usunięcia danych z Bazy Danych:

|  |
| --- |
| Kod |

Aktualizację wykonuje polecenie:

|  |
| --- |
| Kod |

## Biblioteka [nazwa]

Dzięki [Nazwa biblioteki] możemy ułatwić proces operowania na Bazie Danych. Operacje wstawiania danych można zrealizować za pomocą wyrażeń:

|  |
| --- |
| Kod |

Dzięki poniższemu kodu zostanie wykonana operacja usunięcia danych z Bazy Danych:

|  |
| --- |
| Kod |

Aktualizację wykonuje polecenie:

|  |
| --- |
| Kod |

## Biblioteka [nazwa]

Biblioteka [Nazwa] rozwiązuje problem [Trudna funkcjonalność]. Ten fragment kodu realizuje [funkcjonalność]:

|  |
| --- |
| Kod |

# 3. Analiza SWOT

W tej sekcji omawiamy pozytywne i negatywne czynniki pochodzące z wnętrza zespołu lub z zewnątrz:

* Wewnętrzne pozytywne – tzw. Mocne strony
* Wewnętrzne negatywne – tzw. Słabe strony
* Zewnętrzne pozytywne – tzw. Okazje
* Zewnętrzne negatywne – tzw. Zagrożenia

|  |  |  |
| --- | --- | --- |
|  | **Pozytywne czynniki** | **Negatywne czynniki** |
| **Wewnętrzne** | * COBOL nie ma mocnych stron | * Trudna nauka * Stosunkowo wysoki poziom wejścia w AD 2019 * Skomplikowane programy stają się nieczytelne * Brak bibliotek * Brak wiedzy zespołu |
| **Zewnętrzne** | * PM’mi będą mogli się nauczyć (większa moc zespołu) | * Błędy w bibliotekach * Nie wiemy jak wygląda środowisko programowania * Nie wiemy jak wygląda środowisko uruchomieniowe |

# 4. Rekomendacja

Zespół w składzie [Imie Nazwisko] z czystym sumieniem może polecić/odrzucić technologie [Nazwa technologi]. Decyzję motywujemy głównie [czynniki], pomimo zalet w postaci [czynnik].

1. Image taken from Hasso Platter Institute MOOC: Mainframes [link](https://open.hpi.de/courses/mainframes2017) [↑](#footnote-ref-1)