**Sprawozdanie z projektu na PK**

Autor: Rafał Malinowski

12.06.2025

Prowadząca: Dr. Inż. Jolanta Kawulok

Wybrany temat – **7. Drzewo genealogiczne**

# Analiza tematu

Temat można podzielić na dwa osobne zadania – zaprojektowanie sposobu przechowywania i łączenia poszczególnych osób oraz sposobu w jaki chcemy realizować zapytania.

Struktura danych jaka od razu się nasuwa to graf i tak właśnie zostały przechowane dane. Powstała klasa Tree, która przechowuje graf. Każda osoba przechowuje listę swoich dzieci i rodziców, co łączy ze sobą wszystkie węzły.

Trudniejsza część to realizacja zapytań, a przede wszystkim zaprojektowanie dobrej bazy, która pozwoli tworzyć na niej wiele różnych zapytań bez nadmiernej redundancji kodu. Do tego celu zostały stworzone klasy QueryBase oraz Queries. Pierwsza odpowiada za wspomnianą podstawę, a druga dziedzicząc z niej pozwala na prostą implementację zapytań.

# Specyfikacja zewnętrzna

Obsługa programu jest prosta. Wystarczy w linii poleceń podać ścieżkę pliku z strukturą drzewa, pliku z zapytaniami oraz pliku przeznaczonego do przechowania danych wyjściowych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Przykładowe zawartości plików z danymi:

Plik z strukturą drzewa

Lucja Monika

Lucja Damian

Jan Monika

Jan Damian

Kasia Ewa

Kasia Lucja

Kasia Basia

Jurek Ewa

Jurek Lucja

Jurek Basia

Franek Jan

Franek Piotr

Julia Jan

Julia Piotr

Piotr Tomek

Piotr Ida

Piotr Ola

Maria Tomek

Maria Ida

Maria Ola

Iza Franek

Plik z zapytaniami

Monika kuzynostwo

Jan dziadkowie

Franek wnuczkowie

Maria dziadkowie

Franek wnuki

Dane wyjściowe dla przykładowych danych podanych powyżej:

Monika kuzynostwo: Tomek Ida Ola

Jan dziadkowie:

Franek wnuczkowie: Monika Damian Tomek Ida Ola

Maria dziadkowie:

Franek wnuki: Damian Tomek

# Specyfikacja wewnętrzna

**Klasy**:

* Tree – Klasa przechowująca graf drzewa. Jeden obiekt przechowujący drzewo.
* Person – Klasa reprezentująca osobę. Każdy obiekt reprezentuje jedną osobę.
* QueryBase – Podstawa pod zapytania. Jej instancje są tworzone przy tworzeniu obiektów klasy Queries.
* Queries – Przechowuje wszystkie zapytania.
* FileHandler – Odpowiada za otwieranie i zamykanie strumieni plików.
* CliArgs – Obrabia argumenty z linii poleceń oraz częściowo sprawdza ich poprawność.

Jedyna dziedzicząca klasa to Queries, która dziedziczy z QueryBase. Schemat klas:

Obraz zawierający krąg, diagram, Czcionka, szkic

Opis wygenerowany automatycznie

**Istotne pola/metody**

Klasa Queries wykorzystuje elementarne metody klasy QueryBase do składania z nich zapytań. Te kluczowe metody to:

std::shared\_ptr<Person> getMom();

std::shared\_ptr<Person> getDad();

std::vector<std::shared\_ptr<Person>> getParents();

std::vector<std::shared\_ptr<Person>> getChildren();

Takie zapytanie zawsze zwraca wektor wskaźników na osoby, nawet gdy ma zwrócić tylko jeden wskaźnik. Jest to motywowane uproszczeniem logiki, gdyż przypadek zwracania wektora oraz pojedynczego wskaźnika powinny być i tak końcowo obsłużone tak samo.

**Struktura danych**

Drzewo jest reprezentowane za pomocą grafu. Obiekt klasy Tree przechowuje mapę, w której kluczem jest imię osoby a wartością wskaźnik na nią. Każda osoba (instancja klasy Person) posiada pola z wektorami rodziców i dzieci, dzięki którym węzły się łączą.

**Ogólny schemat działania programu**

W pierwszej fazie analizowane są argumenty linii poleceń, z których chwilę później wyciągane są ścieżki plików z danymi oraz pliku wyjściowego. Po ich otwarciu czytany jest plik z strukturą drzewa oraz budowane jest drzewo w pamięci programu. Na bieżąco łączone są ze sobą poszczególne węzły. Po tej fazie czytany jest plik z zapytaniami. Linijka po linijce od razu są one wykonywane. Funkcja wybiera na podstawie tekstu zapytania odpowiednią funkcję zapytania i zwraca wynik, który jest zapisywany do pliku wyjściowego.

# Testowanie i uruchamianie

asd