

14.02.2020

Laboratorium Programowania Komputerów

Temat projektu: walidacja i ekstrakcja danych z numeru PESEL

Autor: Paweł Seweryn

Kierunek: Informatyka

Semestr: 3

Grupa: 4

Sekcja: 1

Prowadzący: Grzegorz Wojciech Kwiatkowski

Link do pobrania projektu:

<https://drive.google.com/open?id=16-PpgFNv8NDKv82ecnHPFhANmIqsPguE>

1. Temat

Zadaniem do zrealizowania w projekcie było stworzenie programu, który na podstawie bazy danych analizuje zawarte w niej numery pesel, sprawdza ich poprawność i ekstrahuje dane tworząc nową bazę i zapisując ją do pliku. W przypadku braku danych wejściowych program powinien wygenerować przykładową bazę o zadanych parametrach.

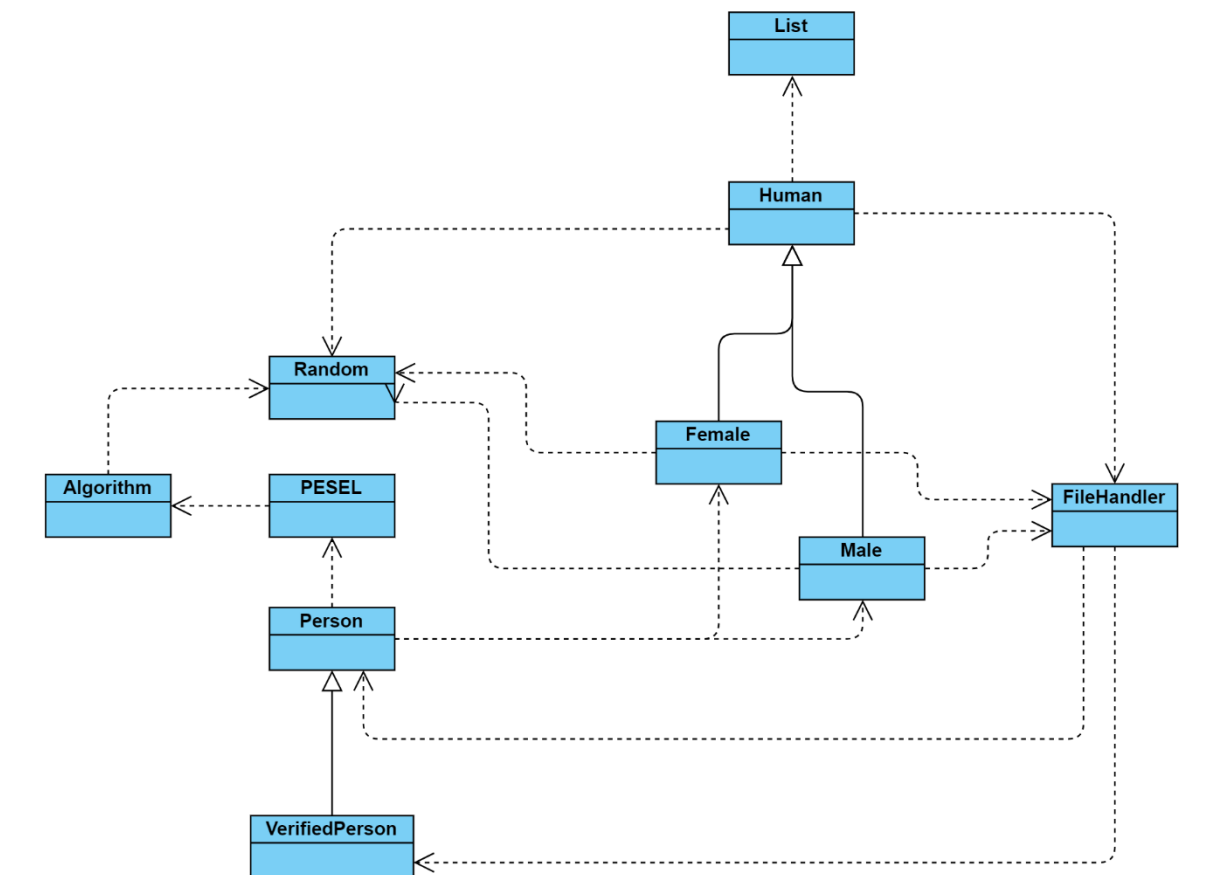
2. Analiza problemu i projektowanie

2.1 Algorytmy i struktury danych

Podstawowym kontenerem dla obiektów jest lista jednokierunkowa, która zaimplementowana została poprzez klasę szablonową „List.h”. Dzięki temu rozwiązaniu listę można zapisać dowolnym typem danych. W programie wykorzystano szereg algorytmów służących m.in. do generowania numerów, weryfikacji i generowania cyfry kontrolnej numeru pesel, porównywania dat.

2.2 Analiza obiektowa

Do realizacji zadania zastosowano obiektowy paradygmat programowania. Stworzono diagram klas, który można zobaczyć poniżej. Linia przerywaną oznaczono zależność, a ciągłą dziedziczenie.



3. Specyfikacja zewnętrzna

Po uruchomieniu programu z konsoli ukazuje się menu tekstowe, w którym dokonuje się wyborów poprzez wpisanie i zatwierdzenie numeru odpowiedniej pozycji. Program może wygenerować bazę danych, utworzyć już istniejącą lub sprawdzić poprawność wprowadzonego numeru pesel.

3.1 Dane wejściowe i komunikaty

Użytkownik może wygenerować bazę danych o zadanej wielkości, nazwie i rozszerzeniu pliku. Formaty plików obsługiwane przez program to txt i csv. Weryfikowana jest poprawność danych wprowadzanych przez konsolę celem uniknięcia błędów – użytkownik informowany jest stosownym komunikatem.

4. Specyfikacja wewnętrzna

4.1 Klasy

Na potrzeby programu napisano 10 klas. Opisane zostały poniżej.

- **Person** – w obiekcie typu tej klasy przechowywane są podstawowe dane rekordu bazy danych takie jak imię i nazwisko, adres i numer pesel
- **VerifiedPerson** – klasa dziedzicząca po *Person*, zawiera dodatkowe informacje takie jak wiek, płeć, datę urodzenia czy poprawność cyfry kontrolnej
- **PESEL** – zawiera tablicę cyfr tworzących numer pesel, a także dzień, miesiąc i rok urodzenia
- **Algorithm** – klasa ta wykorzystywana jest do generowania poprawnego losowego numeru pesel oraz zawiera funkcje umożliwiające sprawdzenie jego poprawności
- **Random** – używana przez wiele innych klas, jej głównym zadaniem jest generowanie losowych liczb całkowitych w zadanym przedziale
- **Human** – klasa używana do generowania losowego adresu i nazwiska
- **Male** – dziedziczy po *Human*, następuje w niej przydzielenie losowego imienia
- **Female** – podobnie jak *Male* dziedziczy po *Human*, oprócz losowego imienia następuje dostosowanie końcówki nazwiska
- **FileHandler** – klasa obsługująca zapis i odczyt plików, generuje listy obiektów
- **List** – klasa szablonowa, jest podstawowym kontenerem na dane używanym w programie (lista jednokierunkowa).

4.2 Funkcje

Poniżej opisane zostały kluczowe funkcje programu.

- `void Algorithm::generateNumber(int day, int month, int year)` – funkcja ta generuje losowo numer pesel o zadanym dniu, miesiącu i roku urodzenia
- `bool Algorithm::isCorrect()` – sprawdza poprawność cyfry kontrolnej numeru pesel
- `void Female::replace(string& str, const string& from, const string& to)` – za pomocą tej funkcji zamienia się końcówki nazwisk na żeńskie

- `List<string> FileHandler::init(const string &fileName)` – funkcja ta jest odpowiedzialna za generowanie list niezbędnych do utworzenia bazy danych
- `bool operator> (const PESEL &p1, const PESEL &p2)` – przeciążony operator porównania. Poprzez datę porównuje dwa numery pesel
- `void List<T>::pushFront(T data)` – funkcja dodaje przekazany jako parametr element na początek listy
- `T List<T>::getData(int number)` – funkcja zwraca obiekt z listy znajdujący się na danym numerze
- `void FileHandler::saveNewDB(const string &fileName, List<VerifiedPerson> &list)` – funkcja zapisuje do pliku wygenerowaną bazę sprawdzonych numerów pesel

Oprócz wyżej wymienionych funkcji program zawiera wiele algorytmów napisanych w konstruktorach, a także mniej znaczących funkcji pomocniczych i przeciążonych operatorach strumienia wyjścia czy konwersji.

5. Testowanie

Programowanie był testowany wielokrotnie zarówno ze względu na poprawność jego działania, jak i efektywność algorytmów. Program dobrze pobierał dane wprowadzone przez użytkownika oraz w razie braku ich poprawności, informował użytkownika stosownym komunikatem.