Projekt "PeselCheck" to aplikacja do weryfikacji numeru PESEL, która sprawdza poprawność numeru pod kątem jego długości oraz wyciąga z niego informacje o płci i cyfrze kontrolnej. Celem projektu było stworzenie programu, który na podstawie wprowadzonego numeru PESEL wyznaczy jego poprawność oraz zidentyfikuje płeć. Do implementacji użyto języka Java, a program uwzględnia algorytm do wyliczania cyfry kontrolnej. System działa poprawnie i spełnia założone wymagania funkcjonalne.

### **Wprowadzenie**

**Cel projektu:**

Celem projektu było stworzenie aplikacji umożliwiającej weryfikację poprawności numeru PESEL oraz wyciąganie z niego dodatkowych informacji, takich jak płeć i cyfra kontrolna.

**Zakres projektu:**

Projekt obejmował implementację funkcji umożliwiających pobieranie numeru PESEL, weryfikację jego poprawności, rozpoznawanie płci oraz obliczanie cyfry kontrolnej.

### **Ustawione wymagania od projektu:**

**Wymagania funkcjonalne:**

* Program powinien przyjmować numer PESEL od użytkownika.
* System musi sprawdzić, czy numer PESEL składa się z 11 cyfr.
* Program powinien rozpoznawać płeć użytkownika na podstawie przedostatniej cyfry.
* System ma obliczać cyfrę kontrolną numeru PESEL i porównywać ją z podaną wartością.

**Technologie i narzędzia:**

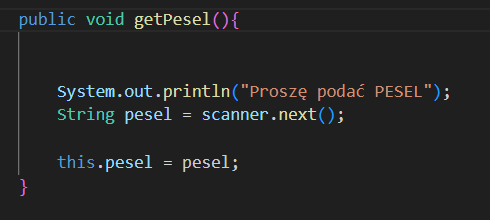
* Język programowania: Java.
* Środowisko programistyczne: IntelliJ IDEA.
* Biblioteki standardowe Javy: Scanner do wprowadzania danych

**Opis implementacji**

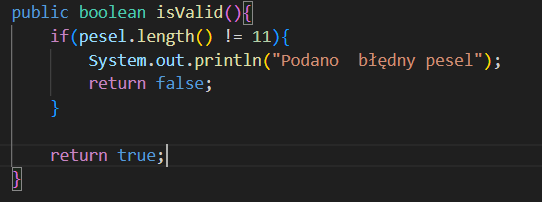
Program składa się z jednej klasy PeselCheck, która zawiera metody do pobierania numeru PESEL, sprawdzania jego poprawności, rozpoznawania płci oraz obliczania cyfry kontrolnej.

**Moduły i komponenty:**

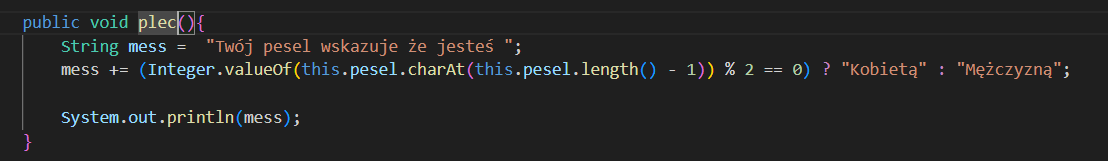
* getPesel(): metoda służąca do wprowadzania numeru PESEL przez użytkownika.



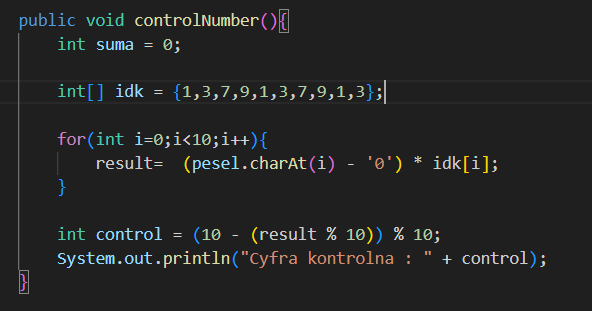
* isValid(): metoda sprawdzająca, czy numer PESEL ma prawidłową długość (11 cyfr).



* plec(): metoda rozpoznająca płeć użytkownika na podstawie przedostatniej cyfry PESEL.



* showControlNumber(): metoda obliczająca cyfrę kontrolną na podstawie algorytmu wagowego.



### **Testowanie**

**Metody testowania:**

Testy jednostkowe zostały przeprowadzone manualnie poprzez wprowadzanie różnych numerów PESEL do programu i weryfikowanie zwracanych wyników.

**Przykłady przypadków testowych:**

* Wprowadzenie prawidłowego numeru PESEL (11 cyfr) i sprawdzenie, czy program poprawnie zwraca płeć oraz cyfrę kontrolną.
* Wprowadzenie zbyt krótkiego numeru PESEL i weryfikacja komunikatu o błędzie.

**Wyniki testów:**

Testy wykazały poprawne działanie programu w przypadku prawidłowego PESEL

### **Rezultaty i wnioski**

**Rezultaty:**

Program spełnia założone cele. Umożliwia poprawną weryfikację numeru PESEL, identyfikację płci oraz obliczenie cyfry kontrolnej.

**Problemy napotkane podczas realizacji:**

Największym wyzwaniem było poprawne zaimplementowanie algorytmu do obliczania cyfry kontrolnej.

### **Podsumowanie**

Projekt zakończył się sukcesem. Stworzono funkcjonalny program weryfikujący numer PESEL. W przyszłości można rozważyć rozwinięcie aplikacji o dodatkowe funkcje lub poprawę interfejsu użytkownika.

### **Streszczenie**

Projekt "numberFile" dotyczy programu służącego do wczytywania, przetwarzania oraz segregowania liczb z pliku tekstowego. Celem projektu było utworzenie aplikacji, która odczyta zestaw liczb z pliku, a następnie podzieli je na liczby parzyste i nieparzyste. Program jest napisany w języku Java. Program skutecznie segreguje dane i wyświetla wynik na konsoli.

### **Wprowadzenie**

**Cel projektu:**

Celem projektu było stworzenie aplikacji umożliwiającej wczytywanie liczb z pliku, a następnie ich sortowanie oraz podział na liczby parzyste i nieparzyste.

**Zakres projektu:**

Projekt obejmował wczytanie liczb z pliku tekstowego, ich sortowanie, rozdzielanie na liczby parzyste i nieparzyste oraz wyświetlenie wyników.

### **Opis wymagań**

**Wymagania funkcjonalne:**

* Program powinien odczytywać dane liczbowe z pliku tekstowego.
* System musi sortować liczby i rozdzielać je na parzyste i nieparzyste.
* Aplikacja powinna wyświetlać na ekranie listę liczb parzystych oraz nieparzystych.

**Technologie i narzędzia:**

* Język programowania: Java
* Narzędzia: BufferedReader do wczytywania pliku, TreeSet do przechowywania i sortowania liczb, technologia TreeSet usuwa duplikaty. Została wykorzystana owa technologia, aby automatycznie posegregować liczby z pliku
* Środowisko: IntelliJ IDEA lub inne kompatybilne IDE obsługujące Javę

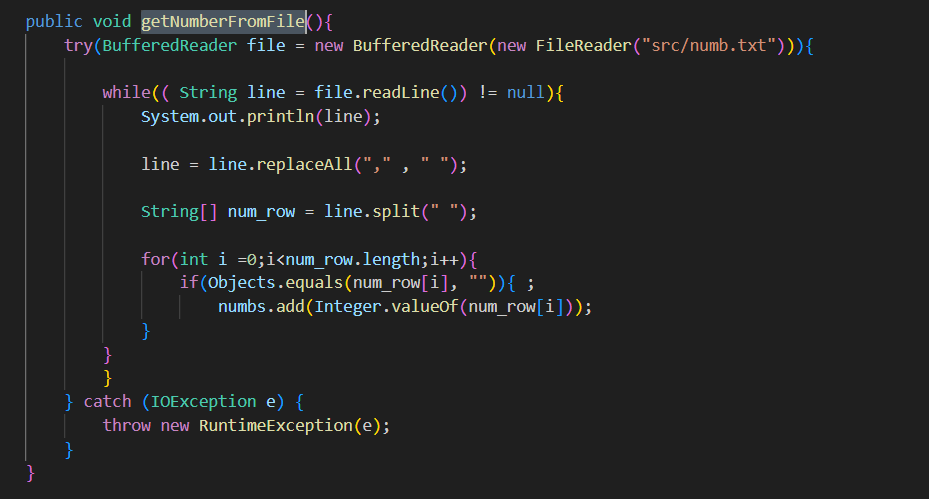
### **Opis implementacji**

**Architektura systemu:**

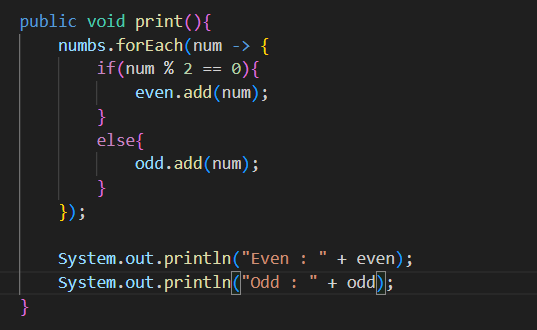
Program jest zbudowany wokół klasy Numbe, która zawiera metody do wczytywania liczb z pliku, ich sortowania oraz podziału na liczby parzyste i nieparzyste.

**Moduły i komponenty:**

* getNumberFromFile(): metoda odpowiedzialna za wczytywanie liczb z pliku tekstowego. Plik ten jest czytany linia po linii, a liczby są konwertowane i dodawane do zbioru numbs.



* print(): metoda dokonująca podziału liczb na parzyste i nieparzyste oraz wyświetlająca wynik.



**Szczegóły implementacji:**

Program używa klasy BufferedReader do wczytywania danych z pliku tekstowego, co zapewnia efektywne i szybkie przetwarzanie pliku. Liczby są przechowywane w obiekcie TreeSet, który automatycznie sortuje liczby i eliminuje duplikaty. Następnie liczby są dzielone na parzyste i nieparzyste

### **Rezultaty i wnioski**

**Rezultaty:**

Program działa poprawnie i spełnia wszystkie założone wymagania. Udało się osiągnąć cel projektu, którym było poprawne wczytywanie, sortowanie i wyświetlanie liczb.

**Problemy napotkane podczas realizacji:**

Głównym wyzwaniem było pobieranie liczb z pliku. Jak również segregowanie liczb, lecz tutaj pomogła mi implementacja interfejsu treeSet.

### **Podsumowanie**

Projekt zakończył się sukcesem. Stworzono aplikację do wczytywania i sortowania liczb z pliku, która spełnia wszystkie wymagania funkcjonalne.