

Programowanie

Projekt zaliczeniowy stworzony na podstawie zadania „PESEL”
(Konkurs „Algorytmion”, rok 2014)

Wiktoria Bronowska

Informatyka III semestr, studia niestacjonarne
Wydział Matematyki Stosowanej

24 luty 2021

Część I

Opis programu

Program został stworzony na podstawie zadania konkursowego „Pesel”. Działanie programu polega na pobraniu od użytkownika informacji o dacie urodzenia, oraz płci użytkownika. Na tej podstawie program generuje indywidualny jedenastocyfrowy numer pesel, przydzielając odpowiednią liczbę początkową, oraz oblicza cyfrę kontrolną. Wszystkie stworzone numery pesel zostają posortowane w kolejności od najstarszego do najmłodszego, a następnie zapisane w nowo utworzonym pliku tekstowym.

Instrukcja obsługi

By móc uruchomić program można użyć odpowiedniego środowiska programistycznego dla języka programowania Java (np. Netbeans), lub dzięki konsoli cmd. Program po wykonaniu działania poinformuje o poprawnym przeniesieniu danych do pliku, oraz zakończy działanie.

Część II

Opis działania

Program polega na wygenerowaniu jedenastocyfrowego numeru PESEL, czyli symbolu jednoznacznie identyfikującego osobę. Podstawowe dane zostają pobrane od użytkownika na poszczególnych etapach działania programu. Do programu zostają wprowadzone wartości zmiennych *birthYear*(rok urodzenia), *birthMonth*(miesiąc urodzenia), *sex*(płeć), *birthDay*(dzień urodzin). Na ich podstawie zostaje wygenerowany numer identyfikacyjny pesel, używając funkcji *peselMethod()*. Program zawiera również metody statyczne takie jak *change()*, która zmienia kolejność występowania elementów w tablicach *years*, *months*, *days*, *ordinalNumTab*. Metoda jest wykorzystywana w innej metodzie statycznej – *sort()*, która sortuje wygenerowane pesele od najstarszego do najmłodszego za pomocą sortowania bąbelkowego. Posortowana tablica zostaje zapisana do pliku tekstowego *pesele.txt*, do czego służy metoda *writeToFile()*, która najpierw sprawdza czy dany plik istnieje a następnie zapisuje dane.

Implementacja

Program działa na podstawie **klasy *PeselCreator***, która zawiera następujące funkcje: *getBirthYear()*, *getBirthMonth()*, *setMonth()*, *getBirthDay()*, *getSex()*, *verification()*, *want()*, *addX()*, *peselMethod()*, *change()*, *sort()*, *wyszukajElemenTablicy()*, *createFile()*, *writeToFile*, oraz funkcja główna *main()*. Do stworzenia programu zostały użyte 7 klas: *java.util.Scanner*, *java.util.LinkedList*, *java.util.List*, *java.util.Random*, *java.io.File*, *java.io.IOException*, *java.io.PrintWriter*.

Działanie funkcji ***getBirthYear()*** polega na pobraniu od użytkownika roku urodzenia z przedziału 1900 i 2299. W sytuacji odczytania liczby wychodzącej poza przedział liczbowy, lub odczytania innej zmiennej niż zmienna liczbową, program poinformuje o trudnościach w przetworzeniu informacji, jeśli następną odczytaną wartością będzie litera „p” program ponownie podejmie próbę odczytania roku urodzenia. Każda inna wartość spowoduje przejście

do funkcji *want()*, która decyduje o dalszym wprowadzaniu danych lub o zakończeniu działania programu. Funkcja kończy działanie zwracając zmienną *birthYear*.

Funkcja *int getBirthYear()*:

```
1      do {
2          try {
3              System.out.println("Podaj rok urodzenia z przedziału 1900 -
4 2299");
5              birthYear = in.nextInt();
6              if (birthYear <= 2299 & 1900 <= birthYear){
7                  break;
8              }
9              else{
10                 System.out.println("Rok musi być liczbą z przedziału
11 1900 - 2299, wciśnij p jeśli chcesz podać ponownie rok, lub wciśnij inny
12 klawisz jeśli chcesz zakończyć");
13                 String n = in.next();
14                 if ("p".equals(n)){
15                     continue;
16                 }
17                 else {
18                     PeselCreator pesel = new PeselCreator();
19                     pesel.want();
20                 }
21             }
22         }
23         catch (Exception e){
24             System.out.println("Podałeś zły format!");
25             System.exit(0);
26         }
27     } while(true) ;
28     return birthYear;
29 }
```

Funkcja ***getBirthMonth()*** polega na pobraniu od użytkownika miesiąca urodzenia. W sytuacji odczytania liczby wychodzącej poza przedział liczbowy (przedział 1 – 12), lub odczytania innej zmiennej niż zmienna liczbową, program poinformuje o trudnościach w przetworzeniu informacji, jeśli następną odczytaną wartością będzie litera „p” program ponownie podejmie próbę odczytania roku urodzenia. Każda inna wartość spowoduje przejście do funkcji *want()*, która decyduje o dalszym wprowadzaniu danych lub o zakończeniu działania programu. Funkcja kończy działanie zwracając zmienną *birthMonth*.

Funkcja *getBirthMonth ()*:

```
1 int getBirthMonth(){
2     int birthMonth;
3     while(true) {
4         try {
5             System.out.println("Podaj miesiąc urodzenia z przedziału 1-
6 12:");
7             birthMonth = in.nextInt();
8             if (birthMonth <= 12 & 1 <= birthMonth){
9                 break;
10            }
11            else{
12                System.out.println("Miesiąc powinien być liczbą z
13 przedziału 1-12, wciśnij p jeśli chcesz podać ponownie miesiąc, lub
14 wciśnij inny klawisz jeśli chcesz zakończyć");
15                String n = in.next();
16                if ("p".equals(n)){
17                    continue;
18                }
19            }
20        }
21        catch (Exception e){
22            System.out.println("Podałeś zły format!");
23            System.exit(0);
24        }
25    } while(true) ;
26    return birthMonth;
27 }
```

```

19         else {
20             PeselCreator pesel = new PeselCreator();
21             pesel.want();
22         }
23     }
24 }
25 catch (Exception e){
26     System.out.println("Podałś złą formę miesiąca!");
27     System.exit(0);
28 }
29 }
30 return birthMonth;
31 }

```

Funkcja **setMonth()** jest funkcją statyczną. Tworzymy w niej zmienną *month*, która przyjmuje taką samą wartość jak *birthMonth*. Na podstawie wartości *birthYear* program ustala jaką liczbą powinna zostać dodana do zmiennej *month*: jeśli rok urodzenia jest liczbą z przedziału 2000 i 2099 – dodajemy 20, jeśli z przedziału 2100 i 2199 – dodajemy 40, jeśli z przedziału 2200 i 2299 – dodajemy 60, a jeśli jest liczbą z przedziału 1900 – 1999 wartość zmiennej *month* pozostaje bez zmian.

Funkcja *setMonth()*:

```

1 int setMonth(int birthYear, int birthMonth){
2     int month = birthMonth;
3     if (2000 <= birthYear & birthYear <= 2099){
4         month += 20;
5     }
6     else if (2100 <= birthYear & birthYear <= 2199){
7         month += 40;
8     }
9     else if (2200 <= birthYear & birthYear <= 2299){
10        month += 60;
11    }
12    return month;
13 }

```

Funkcja **getBirthDay()** tworzy dwie tablice: *oneMonth* przechowującą numery miesięcy z liczbą dni 31, oraz *secondMonth* przechowującą numery miesięcy z liczbą dni 30. Funkcja polega na pobraniu od użytkownika dnia urodzenia. Program sprawdza podaną wcześniej wartość *birthMonth* i zależnie od tego w której tablicy (*oneMonth* czy *secondMonth*) się znajduje, będzie oczekiwał liczby z określonego zakresu. W sytuacji gdy wcześniej podanym miesiącem był luty, program sprawdza czy nie występuje rok przestępny. Jeśli odczytana zmienna nie jest zmienną liczbową, program poinformuje o trudnościach w przetworzeniu informacji, jeśli następną odczytaną wartością będzie litera „p” program ponownie podejmie próbę odczytania roku urodzenia. Każda inna wartość spowoduje przejście do funkcji *want()*, która decyduje o dalszym wprowadzaniu danych lub o zakończeniu działania programu. Funkcja kończy działanie zwracając zmienną *birthDay*.

Funkcja *getBirthDay()*:

```

1 int getBirthDay(int birthYear, int birthMonth){
2     int[] oneMonth = {1, 3, 5, 7, 8, 10, 12};
3     int[] secondMonth = {4, 6, 9, 11};
4     int birthDay;
5     while(true) {
6         try {
7             System.out.println("Podaj dzień urodzenia:");

```

```

8         birthDay = in.nextInt();
9         if (wyszukajElementTablicy(oneMonth, birthMonth) == true) {
10             if (0 < birthDay & birthDay <=31) {
11                 break;
12             }
13             else {
14                 System.out.println("Podales zly numer, wciśnij p
15 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
16 chcesz zakończyć");
17                 String n = in.next();
18                 if ("p".equals(n)){
19                     continue;
20                 }
21                 else {
22                     PeselCreator pesel = new PeselCreator();
23                     pesel.want();
24                 }
25             }
26         }
27         else if (wyszukajElementTablicy(secondMonth, birthMonth)
28 == true) {
29             if (0 < birthDay & birthDay <=31) {
30                 break;
31             }
32             else {
33                 System.out.println("Podales zly numer, wciśnij p
34 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
35 chcesz zakończyć");
36                 String n = in.next();
37                 if ("p".equals(n)){
38                     continue;
39                 }
40                 else {
41                     PeselCreator pesel = new PeselCreator();
42                     pesel.want();
43                 }
44             }
45         }
46         else if (birthMonth == 2){
47             if ((birthYear % 4 == 0) & ((birthYear % 100 != 0) |
48 (birthYear % 400 == 0))) { //leap year
49                 if (0 < birthDay & birthDay <=29) {
50                     break;
51                 }
52                 else {
53                     System.out.println("Podales zly numer, wciśnij p
54 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
55 chcesz zakończyć");
56                     String n = in.next();
57                     if ("p".equals(n)){
58                         continue;
59                     }
60                     else {
61                         PeselCreator pesel = new PeselCreator();
62                         pesel.want();
63                     }
64                 }
65             }
66             else {
67                 if (0 < birthDay & birthDay <=29) {
68                     break;
69                 }
70                 else {

```

```

71         System.out.println("Podales zly numer, wciśnij p
72 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
73 chcesz zakończyć");
74         String n = in.next();
75         if ("p".equals(n)){
76             continue;
77         }
78         else {
79             PeselCreator pesel = new PeselCreator();
80             pesel.want();
81         }
82     }
83 }
84 }
85 }
86 catch (Exception e){
87     System.out.println("Podales złą formę!");
88     System.exit(0);
89 }
90 }
91 return birthDay;
92 }

```

Funkcja **getSex()** polega na pobraniu od użytkownika płci, używając w tym celu metody *linecache.getline()*. Litera „k” oznacza kobietę, a litera „m” mężczyznę, jeśli odczytaną wartością będzie inna litera, lub liczba, program poinformuje o trudnościach w przetworzeniu informacji i zapyta czy chcemy podać ponownie płeć, jeśli następną odczytaną wartością będzie litera „p” program ponownie podejmie próbę odczytania płci. Każda inna wartość spowoduje przejście do funkcji *want()*, która decyduje o dalszym wprowadzaniu danych lub o zakończeniu działania programu. Funkcja kończy działanie zwracając zmienną *sex*.

Funkcja *getSex()*:

```

1 char getSex(){
2     char sex;
3     while(true) {
4         try {
5             System.out.print("Podaj płeć (\"k\" lub \"m\"):");
6             sex = in.next().charAt(0);
7             if (sex == 'm') {
8                 break;
9             }
10            else if (sex == 'k'){
11                break;
12            }
13            else{
14                System.out.println("Podales niepoprawną formę płci,
15 wciśnij p jeśli chcesz podać ponownie płeć, lub wciśnij inny klawisz jeśli
16 chcesz zakończyć");
17                char n = in.next().charAt(0);
18                if (n == 'p'){
19                    continue;
20                }
21                else {
22                    PeselCreator pesel = new PeselCreator();
23                    pesel.want();
24                }
25            }
26        }
27        catch (Exception e){
28            System.out.println("Podales niepoprawną formę płci!");
29            System.exit(0);

```

```

30         }
31     }
32     return sex;
33 }

```

Funkcja **verification()** wczytuje na ekran wszystkie pobrane wcześniej dane i pyta, czy ma dokonać wpisu i wygenerować dla nich pesel. Pobiera od użytkownika literę, jeśli będzie nią „t” (od „tak”), wtedy zostaje uruchamiana funkcja pesel(), która generuje numer pesel, a następnie przechodzi do funkcji want(), pytającej czy chcemy kontynuować program czy go zakończyć.

Funkcja *verification()*

```

1 void verification(int birthYear, int birthDay, int birthMonth, int month,
2 char sex) {
3     System.out.println("Czy chcesz dokonać wpisu: " + birthDay + ", " +
4 birthMonth + ", " + birthYear + ", " + sex + "? Klawisz t - tak, pozostałe
5 - nie:");
6     String n = in.next();
7     if ("t".equals(n)) {
8         PeselCreator pesel = new PeselCreator();
9         pesel.peselMethod(birthYear, month, birthDay, sex);
10        pesel.want();
11    }
12    else {
13        PeselCreator pesel = new PeselCreator();
14        pesel.want();
15    }
16 }

```

Funkcja *want()* pyta czy chcemy dokonać kolejnego wpisu, czy chcemy zakończyć działanie programu.

Funkcja **want ()**

```

1 public String[] want() {
2     System.out.println("Czy chcesz dokonać kolejnego wpisu? Klawisz t -
3 tak, pozostałe - nie:");
4     String m = in.next();
5     String[] sortedPesel = new String[1];
6     if ("t".equals(m)) {
7         PeselCreator pesel = new PeselCreator();
8         pesel.main();
9     }
10    else {
11        PeselCreator pesel = new PeselCreator();
12        sortedPesel = PeselCreator.sort(peselTab, years, months, days,
13 ordinalNumTab);
14        writeToFile(sortedPesel);
15    }
16    return sortedPesel;
17 }

```

Funkcja *addX()* tworzy nową tablicę, taką samą jak tablica *arr[]*, ale powiększoną o wprowadzoną wartość zmiennej *int x*.

Funkcja **addX ()**

```

1 public static int[] addX(int n, int arr[], int x)

```

```

2     {
3         int i;
4         int newarr[] = new int[n + 1];
5         for (i = 0; i < n; i++)
6             newarr[i] = arr[i];
7         newarr[n] = x;
8         return newarr;
9     }

```

Funkcja **peselMethod()** jest odpowiedzialna za wygenerowanie peselu. Pobiera wartości zmiennych *birthYear*, *month*, *birthDay*, *sex* i po odpowiedniej modyfikacji umieszcza je w pustej tablicy *tempPesel*. Dodaje również zmienną *ordinalNumber*, która jest indywidualną liczbą porządkową każdego peselu. Zostaje wygenerowana liczba z przedziału 0 do 9 – nieparzysta dla mężczyzny, parzysta dla kobiety. Następnie program oblicza cyfrę kontrolną wg poniższego wzoru:

$$c = 10 - \left(\sum_{i=1}^{10} c_i w_i \right)_{\text{mod} 10},$$

, gdzie w_i jest i -tą wagą, c_i jest i -tą cyfrą numeru PESEL, $(a)_{\text{mod} b}$ oznacza resztę z dzielenia liczby a przez b . Kolejne wagi wynoszą odpowiednio: 1, 3, 7, 9, 1, 3, 7, 9, 1, 3.

Gdy wszystkie elementy zostaną dodane do tablicy *tempPesel* zostaje ona zamieniona na wartość zmiennej string *res* a następnie wyświetlona w konsoli jako wygenerowany pesel.

Funkcja **peselMethod()**:

```

1 void peselMethod(int birthYear, int month, int birthDay, char sex) {
2     List<Number> tempPesel = new LinkedList<>();
3
4     years = addX(n, years, birthYear);
5     months = addX(n, months, month);
6     days = addX(n, days, birthDay);
7     ordinalNumTab = addX(n, ordinalNumTab, ordinalNumber);
8     tempPesel.add(Math.floorDiv((birthYear % 100), 10));
9     tempPesel.add((birthYear % 100) % 10);
10    tempPesel.add(Math.floorDiv(month, 10));
11    tempPesel.add(month % 10);
12    tempPesel.add(Math.floorDiv(birthDay, 10));
13    tempPesel.add(birthDay % 10);
14    tempPesel.add(Math.floorDiv(ordinalNumber, 100));
15    tempPesel.add(Math.floorDiv(ordinalNumber, 10));
16    tempPesel.add(ordinalNumber % 10);
17    while (true){
18        int numSex = losowanie.nextInt((9)+1);
19        if (numSex % 2 == 0){
20            if ('k' == sex) {
21                tempPesel.add(numSex);
22                break;
23            }
24        }
25        else {
26            if (sex == 'm') {
27                tempPesel.add(numSex);
28                break;
29            }
30        }
31    }
32    while (true) {
33        int sum = 0;
34        for (int i = 0; i < 10; i++){
35            int num = (int) tempPesel.get(i);

```



```

36         int res = wage[i] * num;
37         sum += res;
38     }
39     int control_num = 10 - (sum % 10);
40     if (control_num == 10) {
41         PeselCreator pesel = new PeselCreator();
42         pesel.peselMethod(birthYear, month, birthDay, sex);
43     }
44     else {
45         tempPesel.add(control_num);
46         break;
47     }
48 }
49 String delim = "";
50 StringBuilder sb = new StringBuilder();
51 int i = 0;
52 while (i < tempPesel.size() - 1) {
53     sb.append((int) tempPesel.get(i));
54     sb.append(delim);
55     i++;
56 }
57 sb.append((int) tempPesel.get(i));
58 String res = sb.toString();
59 System.out.println("Wygenerowany pesel:" + res);
60 peselTab = addX(n, peselTab, res);
61 n++;
62 }

```

Funkcja ***change()*** jest metodą statyczną, która zmienia kolejność występowania elementów w tablicach. Jest ona elementem metody *sort()*, sortującą elementy tablicy używając sortowania bąbelkowego.

Funkcja *change()*

```

1 static void change(int j, String[] peselTab, int[] years, int[] months,
2 int[] days, int[] ordinalNumTab){
3     String temp = peselTab[j];
4     peselTab[j] = peselTab[j+1];
5     peselTab[j + 1] = temp;
6     int temp1 = years[j];
7     years[j] = years[j+1];
8     years[j + 1] = temp1;
9     int temp2 = months[j];
10    months[j] = months[j+1];
11    months[j + 1] = temp2;
12    int temp3 = days[j];
13    days[j] = days[j+1];
14    days[j + 1] = temp3;
15    int temp4 = ordinalNumTab[j];
16    ordinalNumTab[j] = ordinalNumTab[j+1];
17    ordinalNumTab[j + 1] = temp4;
18 }

```

Funkcja ***sort()*** jest metodą statyczną, która za pomocą sortowania bąbelkowego sortuje elementy tablicy *peselTab* w kolejności od najstarszego do najmłodszego wykorzystując wcześniej pokazaną funkcję *change()*. Funkcja zwraca tablicę *peselTab*, która zawiera posortowane elementy tablicy zamienione w ciąg znaków string.

Funkcja *sort()*

```

1 static String[] sort(String[] peselTab, int[] years, int[] months, int[]
2 days, int[] ordinalNumTab){
3     int length = peselTab.length;
4     for (int i = 0; i < (length - 1); i++){
5         for (int j = 0; j < (length - i - 1); j++) {
6             if (years[j] < years[j+1]) {
7                 continue;
8             }
9             else if (years[j] > years[j+1]) {
10                PeselCreator.change(j, peselTab, years, months, days,
11 ordinalNumTab);
12            } else {
13                if (months[j] < months[j+1]) {
14                    continue;
15                }
16                else if (months[j] > months[j+1]) {
17                    PeselCreator.change(j, peselTab, years, months,
18 days, ordinalNumTab);
19                }
20                else {
21                    if (days[j] < days[j+1]) {
22                        continue;
23                    }
24                    else if (days[j] > days[j+1]) {
25                        PeselCreator.change(j, peselTab, years, months,
26 days, ordinalNumTab);
27                    }
28                    else {
29                        if (ordinalNumTab[j] < ordinalNumTab[j+1]) {
30                            continue;
31                        }
32                        else {
33                            PeselCreator.change(j, peselTab, years,
34 months, days, ordinalNumTab);
35                        }
36                    }
37                }
38            }
39        }
40    }
41    return peselTab;
42 }

```

Funkcja **wyszukajElementTablicy()** jest metodą , która przeszukuje podaną tablicę *tab[]* i sprawdza czy znajdują się w niej element *int birthMonth*.

Funkcja *wyszukajElementTablicy()*

```

1 boolean wyszukajElementTablicy(int[] tab, int birthMonth) {
2     for (int i = 0; i < tab.length; i++) {
3         if (tab[i] == birthMonth) {
4             return true;
5         }
6     }
7     return false;
8 }

```

Funkcja **createFile()** tworzy plik tekstowy do którego będą zapisywane dane.

Funkcja *createFile()*:

```

1 boolean createFile() {
2     try {

```

```

3     File myObj = new File("pesele.txt");
4     if (myObj.createNewFile()) {
5         System.out.println("File created: " + myObj.getName());
6         return true;
7     } else {
8         System.out.println("File already exists.");
9     }
10    } catch (IOException e) {
11        System.out.println("An error occurred.");
12        e.printStackTrace();
13        return false;
14    }
15        return true;
16 }

```

Funkcja ***writeToFile()*** sprawdza czy istnieje dokument tekstowy *pesele.txt* jeśli tak, to otwiera go i zapisuje w nim wygenerowane pesele. Jeśli dokument nie istnieje, program poinformuje o znalezionym błędzie wyświetlając informację na ekranie konsoli. Po poprawnym zapisaniu danych do pliku zostaje on zamknięty, a użytkownik jest poinformowany o pomyślnym zapisaniu danych.

Funkcja *writeToFile()*

```

1 boolean writeToFile(String[] peselTab) {
2     try
3     {
4         PrintWriter pr = new PrintWriter("pesele");
5
6         for (String peselTab1 : peselTab) {
7             pr.println(peselTab1);
8         }
9         pr.close();
10    }
11    catch (Exception e)
12    {
13        e.printStackTrace();
14        System.out.println("No such file exists.");
15    }
16        return false;
17    }

```

Funkcja ***main()*** odpowiada za uruchomienie wszystkich powyższych funkcji.

```

1 void main() {
2     if (createFile()) {
3         PeselCreator pesel = new PeselCreator();
4         int birthYear = pesel.getBirthYear();
5         int birthMonth = pesel.getBirthMonth();
6         int month = pesel.setMonth(birthYear, birthMonth);
7         int birthDay = pesel.getBirthDay(birthYear, birthMonth);
8         char sex = pesel.getSex();
9         pesel.verifcation(birthYear, birthDay, birthMonth, month, sex);
10        pesel.writeToFile(peselTab);
11    }
12    else {
13        System.out.println("Błąd przy tworzeniu pliku!");
14    }
15 }

```

Testy

Program posiada zabezpieczenia przed wprowadzeniem przez użytkownika niepoprawnych danych. Wprowadzenie wartości zmiennej *birthYear*, *birthMonth*, oraz *birthDay*, która nie jest zmienną liczbową spowoduje pokazanie się na ekranie informacji o błędzie, oraz zapyta czy chcemy wprowadzić dane jeszcze raz czy zakończyć.

Dzięki użytym funkcjom *try... catch*, program potrafi znaleźć błąd przy wprowadzaniu danych, oraz w przypadku otwierania plików sprawdza czy dany plik istnieje.

Pełen kod aplikacji

```
1 package pesel;
2 import java.util.Scanner;
3 import java.util.LinkedList;
4 import java.util.List;
5 import java.util.Random;
6 import java.io.File;
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 public class Pesel {
10     public static void main(String[] args) {
11         PeselCreator p = new PeselCreator();
12         p.main();
13     }
14 }
15 class PeselCreator{
16     public int n = 0;
17     Scanner in = new Scanner(System.in);
18     Random losowanie = new Random();
19     StringBuilder sb = new StringBuilder();
20     public int [] years = new int[n];
21     public int ordinalNumber = 0;
22     int[] wage = {1, 3, 7, 9, 1, 3, 7, 9, 1, 3};
23     public int [] months = new int[n];
24     public int [] days = new int[n];
25     public int [] ordinalNumTab = new int[n];
26     public String [] peselTab = new String[n];
27     private int birthYear;
28
29     int getBirthYear(){
30         do {
31             try {
32                 System.out.println("Podaj rok urodzenia z przedziału 1900
33 - 2299");
34                 birthYear = in.nextInt();
35                 if (birthYear <= 2299 & 1900 <= birthYear){
36                     break;
37                 }
38                 else{
39                     System.out.println("Rok musi być liczbą z przedziału
40 1900 - 2299, wciśnij p jeśli chcesz podać ponownie rok, lub wciśnij inny
41 klawisz jeśli chcesz zakończyć");
42                     String n = in.next();
43                     if ("p".equals(n)){
44                         continue;
45                     }
46                     else {
47                         PeselCreator pesel = new PeselCreator();
48                         pesel.want();
49                     }
50                 }

```

```

51         }
52         catch (Exception e){
53             System.out.println("Podales zly format!");
54             System.exit(0);
55         }
56     } while(true)        ;
57     return birthYear;
58 }
59
60
61 int getBirthMonth(){
62     int birthMonth;
63     while(true) {
64         try {
65             System.out.println("Podaj miesiac urodzenia z przedzialu
66 1-12:");
67             birthMonth = in.nextInt();
68             if (birthMonth <= 12 & 1 <= birthMonth){
69                 break;
70             }
71             else{
72                 System.out.println("Miesiac powinien byc liczba z
73 przedzialu 1-12, wciśnij p jeśli chcesz podać ponownie miesiac, lub
74 wciśnij inny klawisz jeśli chcesz zakończyć");
75                 String n = in.next();
76                 if ("p".equals(n)){
77                     continue;
78                 }
79                 else {
80                     PeselCreator pesel = new PeselCreator();
81                     pesel.want();
82                 }
83             }
84         }
85         catch (Exception e){
86             System.out.println("Podales zla forme miesiaca!");
87             System.exit(0);
88         }
89     }
90     return birthMonth;
91 }
92
93 int setMonth(int birthYear, int birthMonth){
94     int month = birthMonth;
95     if (2000 <= birthYear & birthYear <= 2099){
96         month += 20;
97     }
98     else if (2100 <= birthYear & birthYear <= 2199){
99         month += 40;
100     }
101     else if (2200 <= birthYear & birthYear <= 2299){
102         month += 60;
103     }
104     return month;
105 }
106
107 int getBirthDay(int birthYear, int birthMonth){
108     int[] oneMonth = {1, 3, 5, 7, 8, 10, 12};
109     int[] secondMonth = {4, 6, 9, 11};
110     int birthDay;
111     while(true) {
112         try {
113             System.out.println("Podaj dzien urodzenia:");
114             birthDay = in.nextInt();
115             if (wyszukajElementTablicy(oneMonth, birthMonth) == true)
116 {

```

```

117         if (0 < birthDay & birthDay <=31) {
118             break;
119         }
120     else {
121         System.out.println("Podales zly numer, wciśnij p
122 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
123 chcesz zakończyć");
124         String n = in.next();
125         if ("p".equals(n)){
126             continue;
127         }
128         else {
129             PeselCreator pesel = new PeselCreator();
130             pesel.want();
131         }
132     }
133 }
134 else if (wyszukajElementTablicy(secondMonth, birthMonth)
135 == true) {
136     if (0 < birthDay & birthDay <=31) {
137         break;
138     }
139     else {
140         System.out.println("Podales zly numer, wciśnij p
141 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
142 chcesz zakończyć");
143         String n = in.next();
144         if ("p".equals(n)){
145             continue;
146         }
147         else {
148             PeselCreator pesel = new PeselCreator();
149             pesel.want();
150         }
151     }
152 }
153 else if (birthMonth == 2){
154     if ((birthYear % 4 == 0) & ((birthYear % 100 != 0) |
155 (birthYear % 400 == 0))) { //leap year
156         if (0 < birthDay & birthDay <=29) {
157             break;
158         }
159         else {
160             System.out.println("Podales zly numer, wciśnij p
161 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
162 chcesz zakończyć");
163             String n = in.next();
164             if ("p".equals(n)){
165                 continue;
166             }
167             else {
168                 PeselCreator pesel = new PeselCreator();
169                 pesel.want();
170             }
171         }
172     }
173     else {
174         if (0 < birthDay & birthDay <=29) {
175             break;
176         }
177         else {
178             System.out.println("Podales zly numer, wciśnij p
179 jeśli chcesz podać ponownie numer dnia, lub wciśnij inny klawisz jeśli
180 chcesz zakończyć");
181             String n = in.next();

```

```

182         if ("p".equals(n)){
183             continue;
184         }
185         else {
186             PeselCreator pesel = new PeselCreator();
187             pesel.want();
188         }
189     }
190 }
191 }
192 }
193 catch (Exception e){
194     System.out.println("Podales złą formę!");
195     System.exit(0);
196 }
197 }
198 return birthDay;
199 }
200
201 char getSex(){
202     char sex;
203     while(true) {
204         try {
205             System.out.print("Podaj płeć (\"k\" lub \"m\"):");
206             sex = in.next().charAt(0);
207             if (sex == 'm') {
208                 break;
209             }
210             else if (sex == 'k'){
211                 break;
212             }
213             else{
214                 System.out.println("Podales niepoprawną formę płci,
215 wciśnij p jeśli chcesz podać ponownie płeć, lub wciśnij inny klawisz
216 jeśli chcesz zakończyć");
217                 char n = in.next().charAt(0);
218                 if (n == 'p'){
219                     continue;
220                 }
221                 else {
222                     PeselCreator pesel = new PeselCreator();
223                     pesel.want();
224                 }
225             }
226         }
227         catch (Exception e){
228             System.out.println("Podales niepoprawną formę płci!");
229             System.exit(0);
230         }
231     }
232     return sex;
233 }
234
235 void verification(int birthYear, int birthDay, int birthMonth, int
236 month, char sex) {
237     System.out.println("Czy chcesz dokonać wpisu: " + birthDay + ", "
238 + birthMonth + ", " + birthYear + ", " + sex + "? Klawisz t - tak,
239 pozostałe - nie:");
240     String n = in.next();
241     if ("t".equals(n)){
242         PeselCreator pesel = new PeselCreator();
243         pesel.peselMethod(birthYear, month, birthDay, sex);
244         pesel.want();
245     }
246     else {

```

```

247         PeselCreator pesel = new PeselCreator();
248         pesel.want();
249     }
250 }
251
252 public String[] want() {
253     System.out.println("Czy chcesz dokonać kolejnego wpisu? Klawisz t
254 - tak, pozostałe - nie:");
255     String m = in.next();
256     String[] sortedPesel = new String[1];
257     if ("t".equals(m)) {
258         PeselCreator pesel = new PeselCreator();
259         pesel.main();
260     }
261     else {
262         PeselCreator pesel = new PeselCreator();
263         sortedPesel = PeselCreator.sort(peselTab, years, months,
264 days, ordinalNumTab);
265         writeToFile(sortedPesel);
266     }
267     return sortedPesel;
268 }
269
270 public static int[] addX(int n, int arr[], int x)
271 {
272     int i;
273     int newarr[] = new int[n + 1];
274     for (i = 0; i < n; i++)
275         newarr[i] = arr[i];
276     newarr[n] = x;
277     return newarr;
278 }
279 public static String[] addX(int n, String arr[], String x)
280 {
281     int i;
282     String newarr[] = new String[n + 1];
283     for (i = 0; i < n; i++)
284         newarr[i] = arr[i];
285     newarr[n] = x;
286     return newarr;
287 }
288
289 void peselMethod(int birthYear, int month, int birthDay, char sex) {
290     List<Number> tempPesel = new LinkedList<>();
291
292     years = addX(n, years, birthYear);
293     months = addX(n, months, month);
294     days = addX(n, days, birthDay);
295     ordinalNumTab = addX(n, ordinalNumTab, ordinalNumber);
296     tempPesel.add(Math.floorDiv((birthYear % 100),10));
297     tempPesel.add((birthYear % 100)%10);
298     tempPesel.add(Math.floorDiv(month,10));
299     tempPesel.add(month % 10);
300     tempPesel.add(Math.floorDiv(birthDay,10));
301     tempPesel.add(birthDay % 10);
302     tempPesel.add(Math.floorDiv(ordinalNumber,100));
303     tempPesel.add(Math.floorDiv(ordinalNumber,10));
304     tempPesel.add(ordinalNumber % 10);
305     while (true) {
306         int numSex = losowanie.nextInt((9)+1);
307         if (numSex % 2 == 0) {
308             if ('k' == sex) {
309                 tempPesel.add(numSex);
310                 break;
311             }
312         }

```



```

313         else {
314             if (sex == 'm') {
315                 tempPesel.add(numSex);
316                 break;
317             }
318         }
319     }
320     while (true) {
321         int sum = 0;
322         for (int i = 0; i < 10; i++){
323             int num = (int) tempPesel.get(i);
324             int res = wage[i] * num;
325             sum += res;
326         }
327         int control_num = 10 - (sum % 10);
328         if (control_num == 10) {
329             PeselCreator pesel = new PeselCreator();
330             pesel.peselMethod(birthYear, month, birthDay, sex);
331         }
332         else {
333             tempPesel.add(control_num);
334             break;
335         }
336     }
337     String delim = "";
338     StringBuilder sb = new StringBuilder();
339     int i = 0;
340     while (i < tempPesel.size() - 1) {
341         sb.append((int) tempPesel.get(i));
342         sb.append(delim);
343         i++;
344     }
345     sb.append((int) tempPesel.get(i));
346     String res = sb.toString();
347     System.out.println("Wygenerowany pesel:" + res);
348     peselTab = addX(n, peselTab, res);
349     n++;
350 }
351
352 static void change(int j, String[] peselTab, int[] years, int[]
353 months, int[] days, int[] ordinalNumTab){
354     String temp = peselTab[j];
355     peselTab[j] = peselTab[j+1];
356     peselTab[j + 1] = temp;
357     int temp1 = years[j];
358     years[j] = years[j+1];
359     years[j + 1] = temp1;
360     int temp2 = months[j];
361     months[j] = months[j+1];
362     months[j + 1] = temp2;
363     int temp3 = days[j];
364     days[j] = days[j+1];
365     days[j + 1] = temp3;
366     int temp4 = ordinalNumTab[j];
367     ordinalNumTab[j] = ordinalNumTab[j+1];
368     ordinalNumTab[j + 1] = temp4;
369 }
370
371
372 static String[] sort(String[] peselTab, int[] years, int[] months,
373 int[] days, int[] ordinalNumTab){
374     int length = peselTab.length;
375     for (int i = 0; i < (length - 1); i++){
376         for (int j = 0; j < (length - i - 1); j++) {
377             if (years[j] < years[j+1]) {
378                 continue;

```

```

379         }
380         else if (years[j] > years[j+1]) {
381             PeselCreator.change(j, peselTab, years, months, days,
382 ordinalNumTab);
383         } else {
384             if (months[j] < months[j+1]) {
385                 continue;
386             }
387             else if (months[j] > months[j+1]) {
388                 PeselCreator.change(j, peselTab, years, months,
389 days, ordinalNumTab);
390             }
391             else {
392                 if (days[j] < days[j+1]) {
393                     continue;
394                 }
395                 else if (days[j] > days[j+1]) {
396                     PeselCreator.change(j, peselTab, years,
397 months, days, ordinalNumTab);
398                 }
399                 else {
400                     if (ordinalNumTab[j] < ordinalNumTab[j+1]) {
401                         continue;
402                     }
403                     else {
404                         PeselCreator.change(j, peselTab, years,
405 months, days, ordinalNumTab);
406                     }
407                 }
408             }
409         }
410     }
411 }
412 return peselTab;
413 }
414
415 boolean wyszukajElementTablicy(int[] tab, int birthMonth) {
416     for (int i = 0; i < tab.length; i++) {
417         if (tab[i] == birthMonth) {
418             return true;
419         }
420     }
421     return false;
422 }
423
424 boolean createFile() {
425     try {
426         File myObj = new File("pesele.txt");
427         if (myObj.createNewFile()) {
428             System.out.println("File created: " + myObj.getName());
429             return true;
430         } else {
431             System.out.println("File already exists.");
432         }
433     } catch (IOException e) {
434         System.out.println("An error occurred.");
435         e.printStackTrace();
436         return false;
437     }
438 }
439 return true;
440 }
441
442 boolean writeToFile(String[] peselTab) {
443     try
444     {

```

```

445         PrintWriter pr = new PrintWriter("pesele");
446
447         for (String peselTab1 : peselTab) {
448             pr.println(peselTab1);
449         }
450     pr.close();
451 }
452 catch (Exception e)
453 {
454     e.printStackTrace();
455     System.out.println("No such file exists.");
456 }
457     return false;
458 }
459
460 void main(){
461     if (createFile()){
462         PeselCreator pesel = new PeselCreator();
463         int birthYear = pesel.getBirthYear();
464         int birthMonth = pesel.getBirthMonth();
465         int month = pesel.setMonth(birthYear, birthMonth);
466         int birthDay = pesel.getBirthDay(birthYear, birthMonth);
467         char sex = pesel.getSex();
468         pesel.verifcation(birthYear, birthDay, birthMonth, month, sex);
469         pesel.writeToFile(peselTab);
470     }
471     else {
472         System.out.println("Błąd przy tworzeniu pliku!");
473     }
474 }
475 }

```