POLITECHNIKA POZNAŃSKA
WYDZIAŁ ELEKTRYCZNY
Instytut Automatyki,Robotyki i Inżynierii Informatycznej

Aleksandra Laskowska

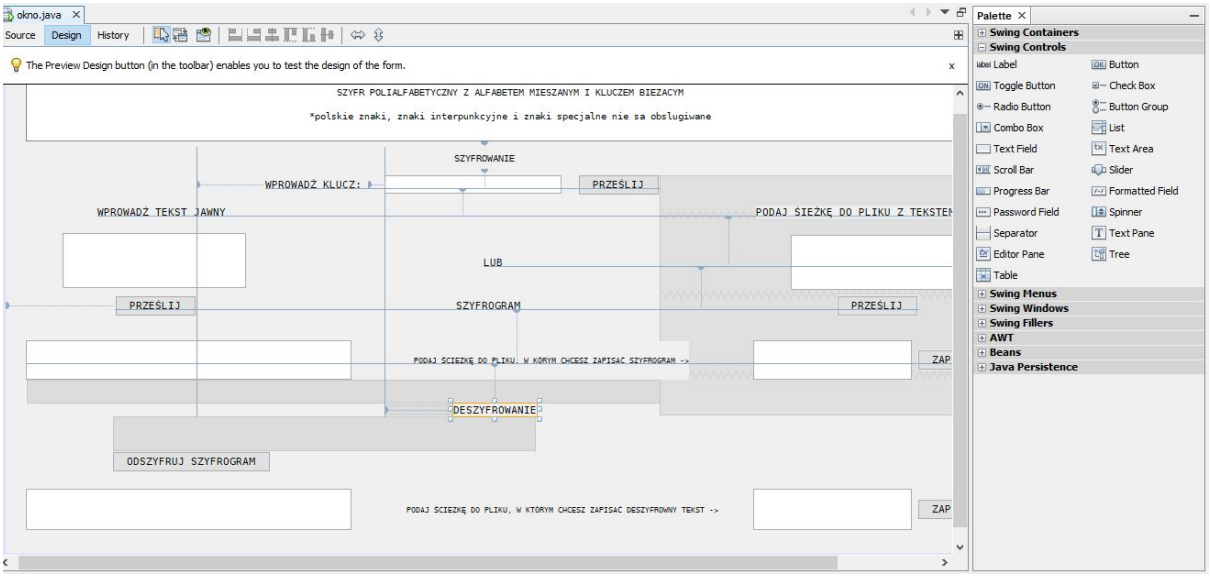
Sprawozdanie z zajęć laboratoryjnych

Implementacja szyfru polialfabetycznego z alfabetem mieszanym i kluczem bieżącym.

18 października 2019 r.

1. Interfejs graficzny

Przy wykonaniu graficznego interfejsu użytkownika (GUI) wykorzystałam swing - bibliotekę graficzną używaną w języku Java. Biblioteka jest bardzo łatwa w obsłudze, ponieważ wybrany element należy wybrać i przesunąć w miejsce, w którym chcemy go zostawić. Później można go edytować zależnie od naszych predyspozycji.



Zrzut 1. "Tworzenie GUI"

2. Implementacja szyfru

Szyfr polialfabetyczny jest systemem z przerywanym przestawieniem kolumnowym. Po wprowadzeniu klucza, jego litery numeruje się w kolejności alfabetycznej. Jeżeli w kluczu litery powtarzają się, nadaje im się kolejny numer. Do przypisania odpowiednich numerów do liter stworzyłam klasę **Para**, której kluczem jest numer litery a wartością dana litera.

```
public class Para {
    public int key;
    public char value;
    Para(int Key, char Value){
        this.key = Key;
        this.value = Value;
    }
   public char getValue(){
       return this.value;
   }

   public int getKey(){
       return this.key;
   }
}
```

Zrzut 2.1 "Klasa Para"

Funkcja **tworzeniePar()** jak mówi jej nazwa, zajmuje się tworzeniem par. Klucz jest przechowywany w dwóch tablicach. Jedną z nich sortuję alfabetycznie, aby później indeksy liter z tablicy uporządkowanej przypisać do liter z tablicy, w której litery są uporządkowane w kolejności ich wprowadzenia. Gdy litery zostaną przyporządkowane jakiemuś indeksowi, zamieniam je na literę spoza alfabetu (w tym przypadku literę ‘ń’), aby nie była przyporządkowana drugi raz.

```
public ArrayList<Para> tworzeniePar (String klucz){
        klucz = klucz.replaceAll(" ", "");

        char [] passTable = klucz.toCharArray();
        char [] passNS = klucz.toCharArray();

        ArrayList<Para> pary = new ArrayList<>();

        // sortowanie tablicy z kluczem by ponumerowac litery w kolejnosci alfabetycznej
        Arrays.sort(passTable);

        //przypisanie numerów do danej litery
      for(int i=0; i<passNS.length;i++){
           for(int j =0; j<passTable.length;j++ ){
               if(passNS[i]==passTable[j]){

               Para temp = new Para((j+1),passNS[i]);
               pary.add(temp);
               passTable[j] = 'ń';
               break;
               }
           }
        }

    return pary;
}
```

Zrzut 2.2 “Funkcja tworzeniePar()”

Kolejnym krokiem jest wybranie numerów przekątnych w prawo i lewo, po których będzie następowało szyfrowanie i deszyfrowanie. Do tego posłużyły mi funkcje **przekatnaP()** oraz **przekatnaL()**. Przekątne powstają w wyniku działania (długość klucza) mod 5. Jeżeli wynik działania jest równy 2 to jest to prawa przekątna, jeżeli 4 - lewa.

```
public ArrayList<Integer> przekatnaP(String klucz){
        int keySize = klucz.length();
        ArrayList<Integer> prawo = new ArrayList<>();
        int mod;
        for (int i =2; i<=keySize; i++){
            mod = i % 5;
            if(mod == 2){
                prawo.add(i);
            }
        }
    return prawo;
}

public ArrayList<Integer> przekatnaL(String klucz){
        int keySize = klucz.length();
        ArrayList<Integer> lewo = new ArrayList<>();
        int mod;
        for (int i =2; i<=keySize; i++){
            mod = i % 5;
            if(mod == 4){
                lewo.add(i);
            }
        }
    return lewo;
}
```

Zrzut 2.3 "Funkcja przekatnaP() i przekatnaL()"

Tekst jawny zapisuję do tablicy dwuwymiarowej. Następnie odczytuję litery z wcześniej wygenerowanych przekątnych idących w prawo lub lewo. Na koniec odczytuję pionowo wszystkie nieodczytane wcześniej litery.

```
x = (int)Math.ceil(dlJawnego/dlKlucza);
y = klucz.length();
k = 0;

M = new char[x][y];
   //zapisanie do macierzy
  for(int i = 0; i<x; i++){
      for(int j = 0; j<y; j++){

          if(k<tekst.length){
             M[i][j] = tekst[k];
          k++;
          }else{
              M[i][j] = '-';
          }
      }
  }
```

Zrzut 2.4 "Zapisanie tekstu jawnego do tablicy dwuwymiarowej"

```
//odczytywanie po prawej przekatnej

for(int i = 0; i < prawo.size(); i++){
    for(int j = 0; j< pary.size(); j++){
        if(prawo.get(i) == pary.get(j).getKey()){
         int l = 0;
            int p = j;
            while(l < x){
              prawaPrzek.add(M[l][p]);
              M[l][p]= '_';
            l++;
            p++;
            if(l==x ||p==y){
                break;
            }
          }


        }
    }
}
```

Zrzut 2.5 "Odczytywanie liter z przekątnych idących w prawą stronę"

```
 //odczytywanie po lewej przekatnej

for(int i = 0; i < lewo.size(); i++){
    for(int j = 0; j< pary.size(); j++){
        if(lewo.get(i) == pary.get(j).getKey()){
         int l = 0;
            int p = j;
            while(l < x){
              lewaPrzek.add(M[l][p]);
              M[l][p]= '_';
            l++;
            p--;
            if(l==x ||p<0){
                break;
            }
          }


        }
    }
}
//usuwanie znakow podkreslenia dla dobrego obliczenia pozycji w koncowej tablicy
for(int i = 0; i< lewaPrzek.size(); i++){
     if(lewaPrzek.get(i).charValue() =='_'){
         lewaPrzek.remove(i);
 }
}
```

Zrzut 2.6 "Odczytywanie liter z przekątnych idących w lewą stronę"

```
//czytanie pionowo

for(int i = 0; i < y; i++){
    for(int j = 0; j < x; j++){

                Pion.add(M[j][i]);
                M[j][i]= '_';

            }
        }



// zlozenie list

szyfr.addAll(prawaPrzek);
szyfr.addAll(lewaPrzek);
szyfr.addAll(Pion);
for(int i = 0; i < szyfr.size(); i++){
    if(szyfr.get(i).charValue() !='_'){
        szyfr2.add(szyfr.get(i));
     }
}
```

Zrzut 2.7 "Odczytywanie liter z idących pionowo"

Lista znaków **szyfr2** zawiera w sobie ostateczną formę szyfrogramu.

Aby deszyfrować tekst z listy **szyfr2** odczytuję litery idące najpierw po prawej, później lewej przekątnej. Na koniec odczytuję znaki idące pionowo w celu odtworzenia tablicy identycznej do pierwotnej tablicy dwuwymiarowej z tekstem jawnym.

```
      //odszyfrowanie po prawej przekatnej
for(int m = 0; m < prawo.size(); m++){
          for(int n = 0; n< pary.size(); n++){
              if(prawo.get(m) == pary.get(n).getKey()){
              int l = 0;
              int p = n;
              while(l < x){
                if(s<prawo.size()*x){
                  O[l][p] = szyfr.get(s);

                }else{
                    O[l][p] = '^';
                }
                  s++;
               l++;
               p++;
              if(l==x ||p==y){
                break;
                  }
              }


              }
          }
        }
```

Zrzut 2.8 "Deszyfrowanie po prawej przekątnej"

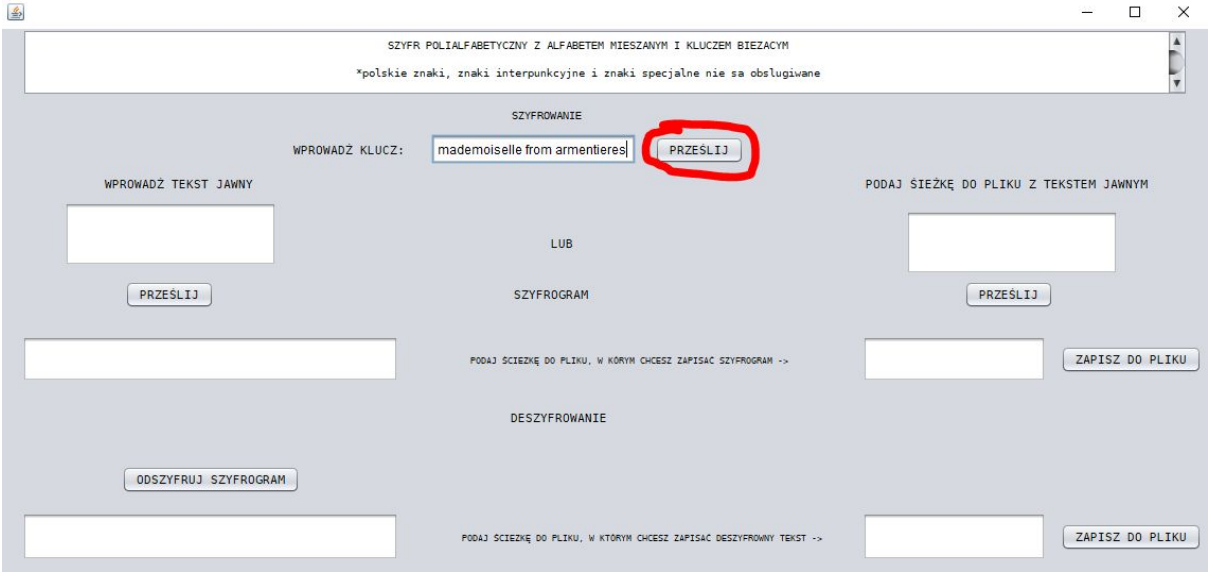
```
//odszyfrowanie po lewej przekatnej

for(int i = 0; i < lewo.size(); i++){
    for(int j = 0; j< pary.size(); j++){
        if(lewo.get(i) == pary.get(j).getKey()){
         int l = 0;
            int p = j;
            while(l < x){
              if(s<lewo.size()*x+prawo.size()*x && O[l][p] == '?'){
                 O[l][p] = szyfr.get(s);
                 s++;
            l++;
            p--;

              }else{
                  l++;
                  p--;
              }

            if(l==x ||p<0){
                break;
            }
         }

        }
    }
}
```

Zrzut 2.9 "Deszyfrowanie po lewej przekątnej"

```
//odszyfrowanie pionowo

 for(int i = 0; i < y; i++){
    for(int j = 0; j < x; j++){

            if(s<szyfr.size()&& O[j][i] == '?'){
                O[j][i] = szyfr.get(s);
                s++;
                 }

           }
         }

  for(int i =0; i<x ; i++){
     for (int j=0; j<y; j++){
     odszyfrowane.add(O[i][j]);
 }
}
```

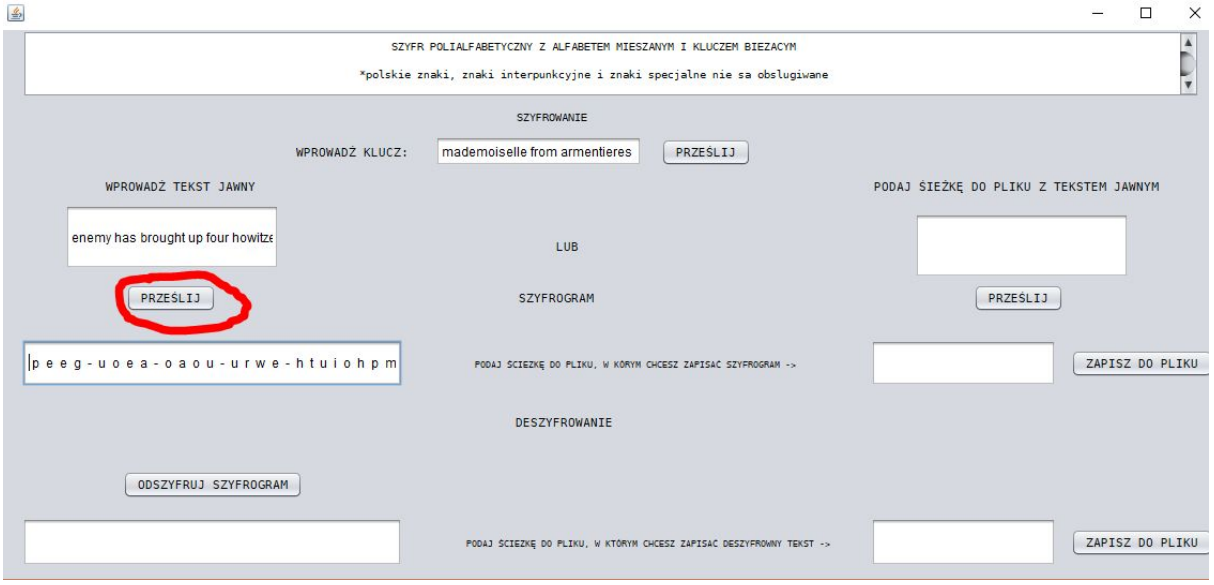
Zrzut 2.10 "Deszyfrowanie pionowo"

3. Obsługa programu



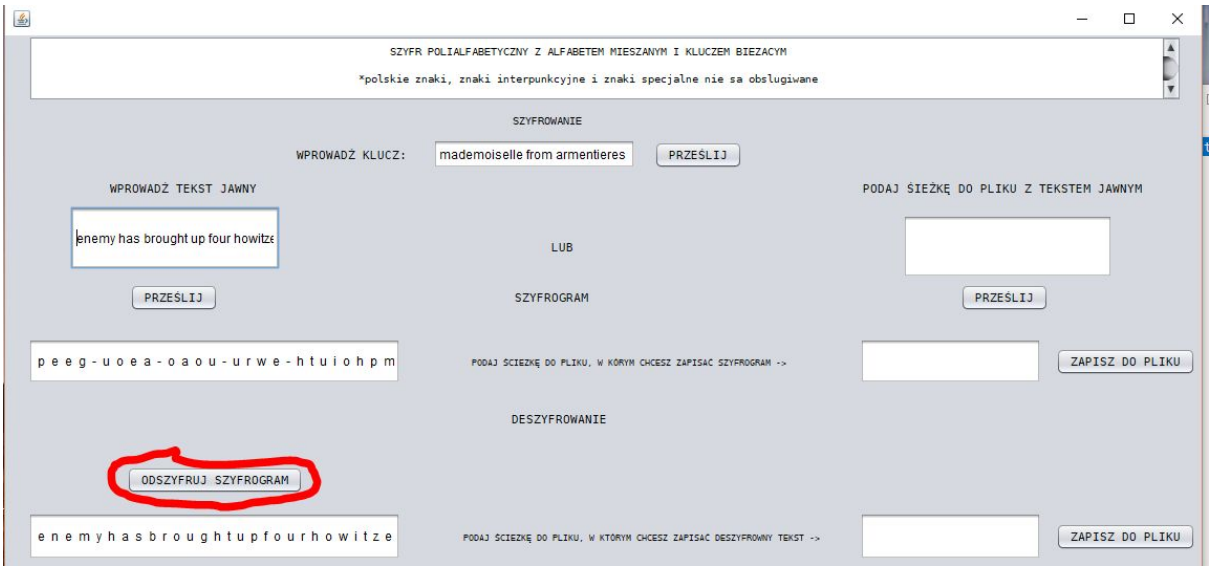
Zrzut 3.1 "Wprowadzenie klucza"

Pierwszym krokiem w obsłudze programu jest wprowadzenie klucza. Aby to zrobić należy wpisać klucz w okienko i nacisnąć zaznaczony przycisk.



Zrzut 3.2 "Wprowadzenie tekstu jawnego i szyfrowanie"

Następnym krokiem jest wprowadzenie tekstu jawnego, kiedy już to zrobimy, naciskamy zaznaczony przycisk a szyfrogram zostanie automatycznie wygenerowany. Aby zapisać szyfrogram, w okienku obok należy wpisać ścieżkę do pliku, w którym chcemy zapisać szyfrogram i nacisnąć przycisk 'ZAPISZ DO PLIKU'.



Zrzut 3.3 "Deszyfrowanie"

Aby deszyfrować tekst należy nacisnąć zaznaczony wyżej przycisk. W okienku niżej pokaże się odszyfrowany szyfrogram. Tekst można zapisać do pliku, którego ścieżkę podamy w okienku obok i naciśniemy przycisk 'ZAPISZ DO PLIKU'.