

VILNIAUS UNIVERSITETAS  
MATEMATIKOS IR INFORMATIKOS FAKULTETAS  
PROGRAMŲ SISTEMOS

**Kodavimo teorija**  
**Golėjaus (Golay) kodas  $C_{23}$**   
Pijus Zlatkus

Vilnius  
2023

# **Turinys**

<b>1. Įvadas .....</b>	<b>3</b>
<b>2. Realizuotos programos dalys .....</b>	<b>3</b>
<b>2.1. Vartotojo pusės realizuotos dalys .....</b>	<b>3</b>
<b>2.2. Programuotojo pusės realizuotos dalys .....</b>	<b>3</b>
<b>3. Naudotos bibliotekos.....</b>	<b>3</b>
<b>4. Užtruktas laikas .....</b>	<b>4</b>
<b>4.1. Užtruktas laikas etapais.....</b>	<b>4</b>
<b>4.2. Visas užtruktas laikas .....</b>	<b>4</b>
<b>5. Programos paleidimas .....</b>	<b>4</b>
<b>6. Programos failai .....</b>	<b>5</b>
<b>6.1. Failų išdėstymas ir aprašymas .....</b>	<b>5</b>
<b>7. Vartotojo sąsaja .....</b>	<b>5</b>
<b>7.1. Pagrindinis menu .....</b>	<b>5</b>
<b>7.2. Scenarijaus vykdymas .....</b>	<b>6</b>
7.2.1. Pirmas scenarijus (vektoriaus kodavimas) .....	6
7.2.2. Antras scenarijus (teksto kodavimas) .....	7
7.2.3. Trečias scenarijus (paveikslėlio kodavimas) .....	8
<b>8. Padaryti programiniai sprendimai.....</b>	<b>9</b>
<b>9. Eksperimentas .....</b>	<b>10</b>
<b>10. Literatūra .....</b>	<b>11</b>

## 1. Įvadas

Programa atlieka pilną Golėjaus (Golay) kodo panaudojimą ir informacijos juo užkodavimą. Užduotis yra atlikta naudojant C# kalbą, nenaudojant jokių papildomų nestandartinių bibliotekų.

## 2. Realizuotos programos dalys

Pagal reikalavimus yra realizuotos visos programos dalys iš vartotojo ir programuotojo pusės.

### 2.1. Vartotojo pusės realizuotos dalys

Programoje yra realizuoti visi trys scenarijai:

- Vektoriaus kodavimas, jo persiuntimas kanalu, atvaizdavimas, galimybė jį keisti ir dekodavimas
- Teksto siuntimas kanalu ir atvaizdavimas naudojant ir nenaudojant kodavimo
- BMP formato paveikslėlio siuntimas kanalu ir atvaizdavimas sukuriant 2 naujus paveikslėlius, kurie parodo kaip skiriasi naudojant ir nenaudojant kodavimą

Kadangi papildomų parametrų šiai užduočiai nėra, tai yra tik leidžiama nurodyti klaidos tikimybę ir norimą užkoduoti informaciją.

### 2.2. Programuotojo pusės realizuotos dalys

Programoje yra realizuoti visi šie scenarijai:

- Baigtinis kūnas  $F_q$ , veiksmas su jo elementais: sudėtis, daugyba ir t.t.
- Veiksmas su matricomis, sudarytomis iš baigtinio kūno  $F_q$  elementų.
- Kodavimas kodu  $C$ .
- Siuntimas kanalu.
- Dekodavimas, naudojant nurodytą dekodavimo algoritmą iš duotos literatūros
- Teksto (paveiksluko) skaidymas reikiamo ilgio vektoriais iš kūno  $F_q$  elementų. Vektorių iš kūno  $F_q$  elementų sujungimas į vieną tekstą (paveiksluką).

## 3. Naudotos bibliotekos

Šiai programai buvo naudotos tik standartinės .NET bibliotekos skirtos teksto atvaizdavimui konsolėje, duomenų tipams ir paveikslėlio nuskaitymui.

## 4. Užtruktas laikas

### 4.1. Užtruktas laikas etapais

- Literatūros skaitymas: **0,5 val.**
- Kodo veikimo aiškinimasis: **1 val.**
- Projektavimas: **1,5 val.**
- Programavimas: **7 val.**
- Klaidų ieškojimas ir taisymas: **1 val.**
- Ataskaitos ruošimas: **1 val.**

### 4.2. Visas užtruktas laikas

Iš viso buvo užtrukta laiko: **12 val.**

## 5. Programos paleidimas

Sukompiliuotą ir paruoštą paleisti programą galima paleisti per **UI.exe** failą, kuris randasi **Release** aplanke. Pats programos kodas yra **GolayCoding** aplanke kurį galima testuoti ir keisti per Visual Studio ar kitas programas atidarant **GolayCoding.sln**. Paleidžiant jokių parametrų nereikia nurodyti, kadangi tai jau yra vykdomasis failas (parametrai yra įvedami jau paleidus programą).

## 6. Programos failai

Visi programos failai kuriuos galima peržiūrėti yra po **GolayCoding** aplanku.

### 6.1. Failų išdėstymas ir aprašymas

- Vartotojo sąsaja ir pagrindinis failas: *GolayCoding/UI/Program.cs*
- Golėjaus kodavimo failas, kuriame yra aprašytos naudojamos matrixos, pagrindinės kodavimo funkcijos, kurios dirba su 3 skirtingais scenarijais su vektoriumi, tekstu ir paveikslėliais: *GolayCoding/GolayCode/Golay.cs*
- Užkodavimo funkcijos failas: *GolayCoding/GolayCode/EncodingService.cs*
- Kanalo funkcijos failas: *GolayCoding/GolayCode/ChannelService.cs*
- Dekodavimo funkcijos ir su ja susijusių kitų funkcijų failas: *GolayCoding/GolayCode/DecodingService.cs*
- Pagalbinių konvertavimo funkcijų, kurių reikia sklandžiam veikimui, failas: *GolayCoding/GolayCode/Extensions.cs*
- Kodavimui naudojamo specialaus klaidos tipo failas: *GolayCoding/GolayCode/GolayException.cs*

## 7. Vartotojo sąsaja

### 7.1. Pagrindinis menu

Vartotojo sąsaja sudaryta iš pagrindinio menu, kuriame galima pasirinkti vieną iš scenarijų

```
Golay Code program

Coding scenarios:
1. Vector:
2. Text
3. BMP Image
Choose scenario by typing a number [1-3]:
|
```

Kad pasirinkti vieną iš scenarijų, reikia parašyta skaičių nuo 1 iki 3. Kitu atveju bus išmesta klaida.

```
Choose scenario by typing a number [1-3]:  
abc  
Incorrect option. Try again.
```

## 7.2. Scenarijaus vykdymas

Pasirinkus vieną iš scenarijų, yra prašoma įvesti klaidos tikimybę, kuri bus naudojama kanale. Tikimybę turi būti daugiau nei 0 ir mažiau nei 1. Rašant skaičių reikia naudoti kablelį, o ne tašką.

```
Golay Code program  
Coding vector  
  
Write a failure probality from 0 to 1:  
0,05|
```

Neteisingai įvedus skaičių yra išmetama klaida.

```
0.05  
Failure probability must be from 0 to 1 (e.g 0,005). Try again.
```

### 7.2.1. Pirmas scenarijus (vektoriaus kodavimas)

Įrašius teisingai tikimybę yra prašoma įrašyti 12 ilgio vektorių (skaičių eilę sudarytą iš 1 ir 0). Neatitikus ilgio ar parašius kitą simbolį nei 0 ar 1 yra išmetama klaida.

```
Write a 12 length vector [from 0 and 1] to code:  
00001111000  
Incorrect vector. Try again.  
000011110000|
```

Pateikus tinkamą vektorių yra parodomas naujas langas su pradiniais duomenimis, užkoduotu vektoriumi ir vektoriumi kuriame yra parodytos klaidos po užkodavimo ir persiuntimo kanalu.

```
Golay Code program  
Starting coding...  
Vector is: 000011110000  
Failure probability is: 0,2  
  
Encoded vector is: 0000111100000111010111  
After sending through noisy channel (4 errors in red): 00101111100001111101101  
Do you want to change vector before decoding it? [y/n] |
```

Po to vartotojas yra paklausiamas ar nori pakeisti vektorių prieš jį dekoduojant. Jei taip, tada yra vėl prašoma įvesti 23 ilgio vektorių.

```
Golay Code program
Starting coding...
Vector is: 000011110000
Failure probability is: 0,2

Encoded vector is: 00001111000001110101111
After sending through noisy channel (4 errors in red): 0010111110000111101101
Do you want to change vector before decoding it? [y/n] y
Write a new 23 length vector: 11110000111100001111000
Decoded vector is: 111100001111
```

Galiausiai programa dekoduoja ir parodo galutinį vektorių.

### 7.2.2. Antras scenarijus (teksto kodavimas)

Šis scenarijus yra paprastesnis, nes užtenka pateikti netuščią sakinio eilutę.

```
Golay Code program
Coding text

Write a failure probability from 0 to 1:
0,05
Write a not empty string to code:
Tekstas, kuri programa uzkoduos|
```

Pateikus tekstą jis yra atvaizduojamas persiunčiant kanalu nenaudojant kodavimo ir persiunčiant kanalu naudojant kodavimą.

```
Golay Code program
Starting coding...
String is: Tekstas, kuri programa uzkoduos
Failure probability is: 0,05

After sending string through noisy channel:
String without coding: Tekstms, c5ri?psog2am' urko?u?s
String with Golay coding: Tekstas, kuri pxograma uzkoduos
```

### 7.2.3. Trečias scenarijus (paveikslėlio kodavimas)

Šiam scenarijui reikia nurodyti teisingą pilną BMP formato paveikslėlio adresą kompiuteryje. Jei failas neegzistuos ar bus netinkamas formatas, bus išvesta klaida.



```
Golay Code program
Coding BMP image

Write a failure probability from 0 to 1:
0,01
Write a full path to BMP image to code:
D:\blackbuck.bmp|
```

Nurodžius teisingai failą, jis yra nuskaitomas ir kaip su tekstu yra sukuriama 2 paveikslėliai iš kurių vienam buvo naudotas kodavimas, o kitam ne.

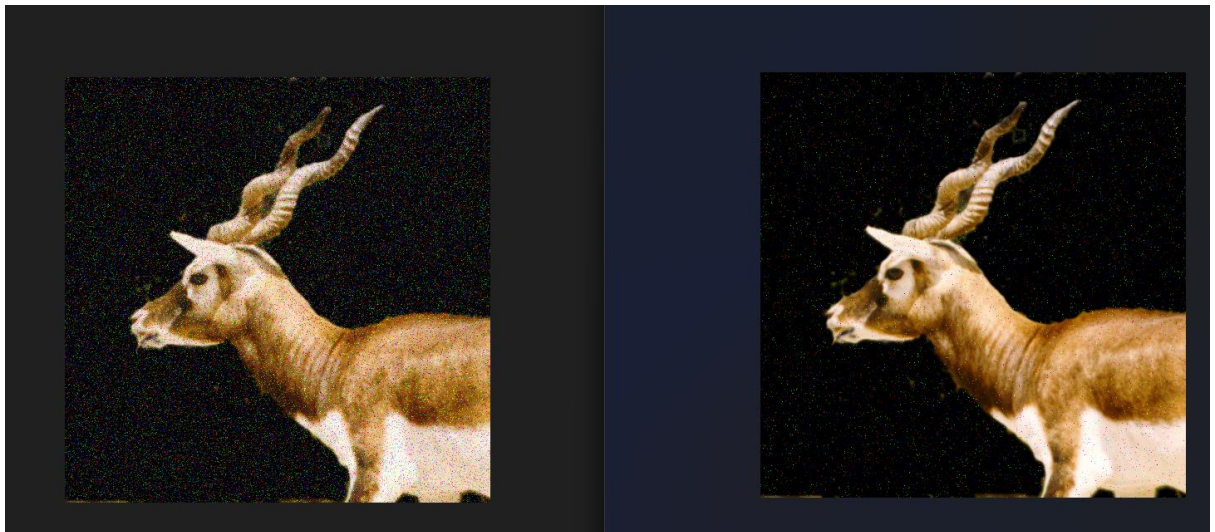
```
Golay Code program
Starting coding...
Processing image without coding...
Processing image with coding...
New files are created
```

Failai atsiranda tame pačiame **Release** aplanke jei paleidžiama per vykdomąjį failą. Jei paleidžiama per Visual Studio, tai nuotraukos atsiranda *GolayCoding/UI/bin/Debug/net8.0* aplanke.

 image_with_coding.bmp	2023-12-06 23:08	BMP File	769 KB
 image_without_coding.bmp	2023-12-06 23:08	BMP File	769 KB



Va kaip atrodo paveikslėliai nenaudojant ir naudojant kodavimą.



## 8. Padaryti programiniai sprendimai

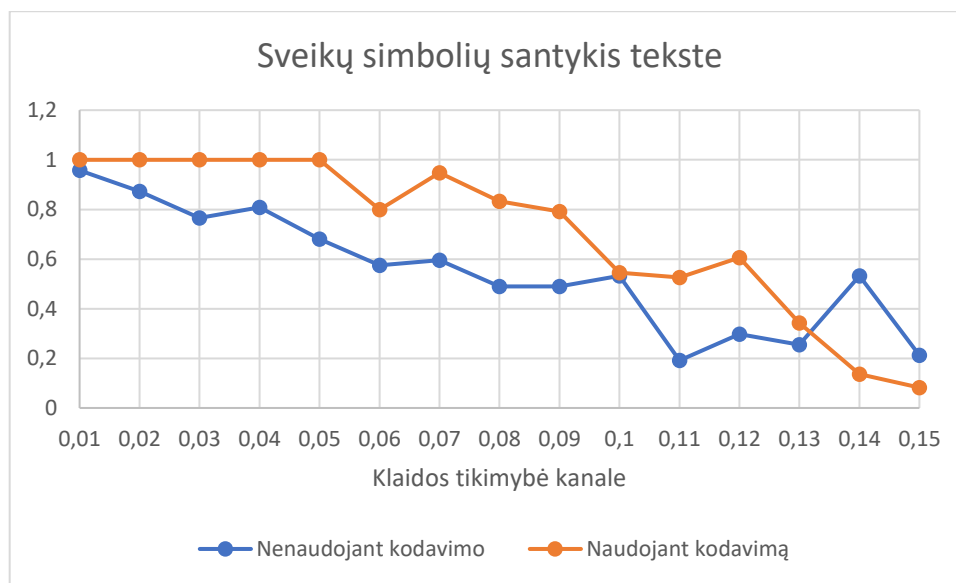
- Vektorius, visą tekstą ir paveikslėlius nuspręsta paversti į bitų masyvo tipą (BitArray), kuris atvaizduoja kiekvieną bitą atskirai
- Keičiant teksto ar paveikslėlio baitus į bitus yra pridedami dar 4 tušti (nuliai) bitai, kadangi baitas yra sudarytas iš 8 bitų, o kodavimui reikia 12
- Kiekvienas baitas yra užkoduojamas, siunčiamas kanalu ir dekoduojamas atskirai
- Pagrindinis vartotojas sąsajos failas vykdo tik scenarijaus pasirinkimo ir tikimybės įvedimo funkcionalumą. Kiekvieno scenarijaus vykdymo funkcijos yra aprašytos pagrindiniame kodavimo *Golay.cs faile*

## 9. Eksperimentas

Kaip eksperimentą pasirinkau patikrinti šio algoritmo klaidų taisymo efektyvumą tekste. Tai yra kaip ištaisytų klaidų (simbolių) kiekis keičiasi nuo kanalo klaidų tikimybės. Kadangi naudojant tikimybes ne visada yra padaroma vienodas klaidų skaičius, tai jų ištaisymą matavau santykinai. Testavimui atlikti naudojau programos antrą scenarijų.

Eksperimentui naudojamas tekstas: „Tekstas, kuri Golejaus kodas turi istaisyti.“

Kanalo klaidos tikimybė keičiasi nuo 0,01 iki 0,15 keičiant po 0,01. Iš viso buvo atlikta 15 testų.



Pagal gautus duomenis galima pastebėti, kad kanalo klaidų tikimybei nedaug didėjant labai keičiasi ištaisytų klaidų skaičius. Tačiau diagrama įrodo, kad naudojant kodavimą išlaikoma didelė dalis nesugadinto teksto.

## 10. Literatūra

<https://klevas.mif.vu.lt/~skersys/doc/ktkt/literatura12.pdf>

A13 Užduoties aptarimas