

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
PROGRAMŲ SISTEMŲ KATEDRA

Kodavimo teorijos A14 laboratorinis darbas

Atliko: 4 kurso 5 grupės studentas
Aurelijus Banelis

Vilnius – 2012

Turinys

1. Užduotis.....	2
2. Programos naudojimas.....	2
2.1. Kompiliavimas.....	2
2.2. Paleidimas.....	2
3. Programa naudotojo požiūriu.....	3
4. Programa programuotojo požiūriu.....	4

1. Užduotis

Duotas kodas C virš Fq. Modeliuoti jo veikimą:

- užkoduoti informaciją,
- siųsti ją nepatikimu kanalu, t.y. su duota tikimybe pe atsitiktinai joje padaryti klaidų,
- dekoduoti, naudojant nurodytą dekodavimo algoritmą iš duotos literatūros.

Parametrai:

- Kūnas: $q=2$
- Kodas: sąšukos (konvoliucinis, *convolutional*) kodas iš [Ber84, Pvz. 15.62, p. 389]
- Dekodavimo algoritmas: dekodavimas su grįžtamuuoju ryšiu (*feedback decoder*) (§15.63)

2. Programos naudojimas

2.1. Kompiliavimas

Programą galima sukompiliuoti naudojant *Java Development Kit* per komandinę eilutę:

```
javac -encoding utf8 -d dist/ src/lt/banelis/aurelijus/*.java  
src/lt/banelis/aurelijus/data/*.java src/lt/banelis/aurelijus/connectors/*.java  
jar cvfm AurelijusA14.jar MANIFEST.MF -C dist/ .
```

Jei *javac* arba *jar* nebus rasti, tai juos reikia pakeisti į pilna adresą iki JDK programų. Pavyzdžiui:

- *javac* → *c:\program files\java\jdk1.6.0_16\bin\javac.exe*
- *jar* → *c:\program files\java\jdk1.6.0_16\bin\jar.exe*

Kataloge *dist* bus laikomi sukompiliuoti *.class* failai. MANIFEST.MF nurodo, kurią klasę paleisti.

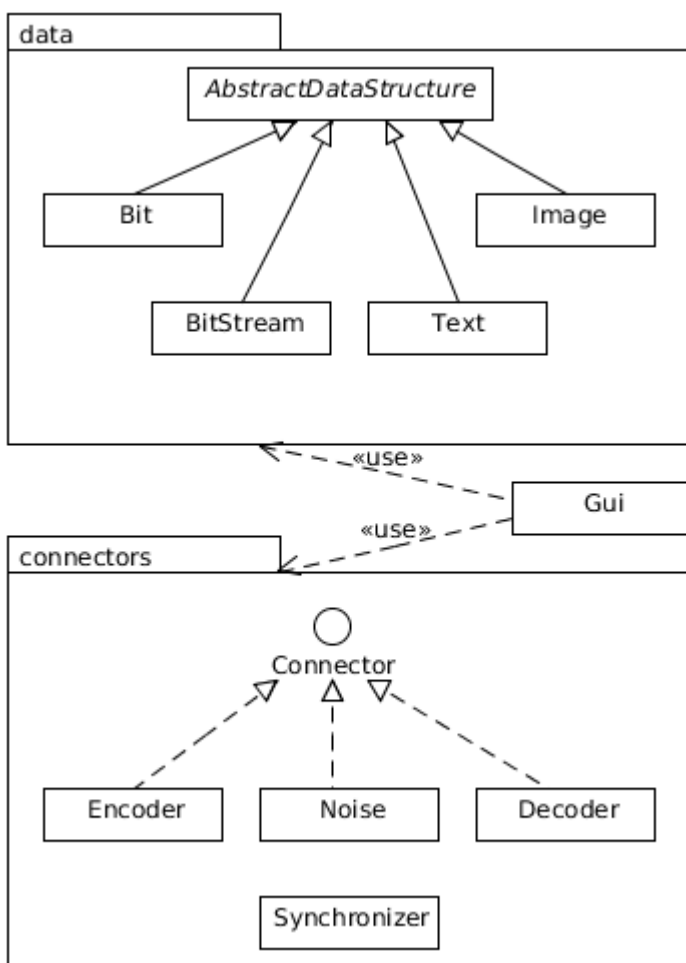
2.2. Paleidimas

Programa parašyta Java kalba, todėl ją paleisti reikėtų iš komandinės eilutės:

```
java -jar AurelijusA14.jar
```

Arba (jei operacinė sistema atpažįsta failą), *AurelijusA14.jar* failą atsidaryti su *Java Platform SE binary* ar analogišku Java interpretatoriumi.

4. Programa programuotojo požiūriu



Programą sudaro du pagrindiniai paketai:

1. *data* – skirtas įvairių duomenų tipų (pvz. bitų, teksto, paveikslėlių) įvedimo, saugojimo ir atvaizdavimo realizavimui.
2. *connectors* – skirtas įvairioms duomenų transformacijoms (pvz. kodavimui, triukšmams ar dekodavimui).

Gui klasė skirta programos paleidimui, ji sudėlioja visus elementus į grafinę naudotojo sąsają. Funkcijomis *encode()*, *transfer()* ir *decode()* atliekamas¹ įvestų duomenų užkodavimas, persiuntimas kanalu ir dekodavimas.

Kiekvienas *data* ir *connectors* paketo klasių kodas organizuojamas tokiu principu:

- Failo pradžioje aprašomos skaičiavimui ir saugojimui naudojamos funkcijos.
- Failo pabaigoje aprašomos grafinei naudotojo sąsajai skirtos funkcijos (pvz. teksto įvedimo laukelių sukūrimas arba registrų reikšmių atnaujinimas).

Synchronizer klasė yra skirta siuntėjo-gavėjo duomenų palyginimui, nes dėl dekodavimo naudojant registrus susidaro 6 bitų vėlavimas.

Katalogas *test* skirtas automatiniais *JUnit* testams atlikti. Šioje programoje jie naudoti patikrinti teksto ir paveikslėlio vertimo į ir iš dvejetainio formato teisingumo patikrinimui.

¹ Minėtos funkcijos iškviečia jau atskirų klasių (pvz. *Encoder*, *Text*, *Decoder*) paprogrames.