Csontó Benjámin
JTB4Y1

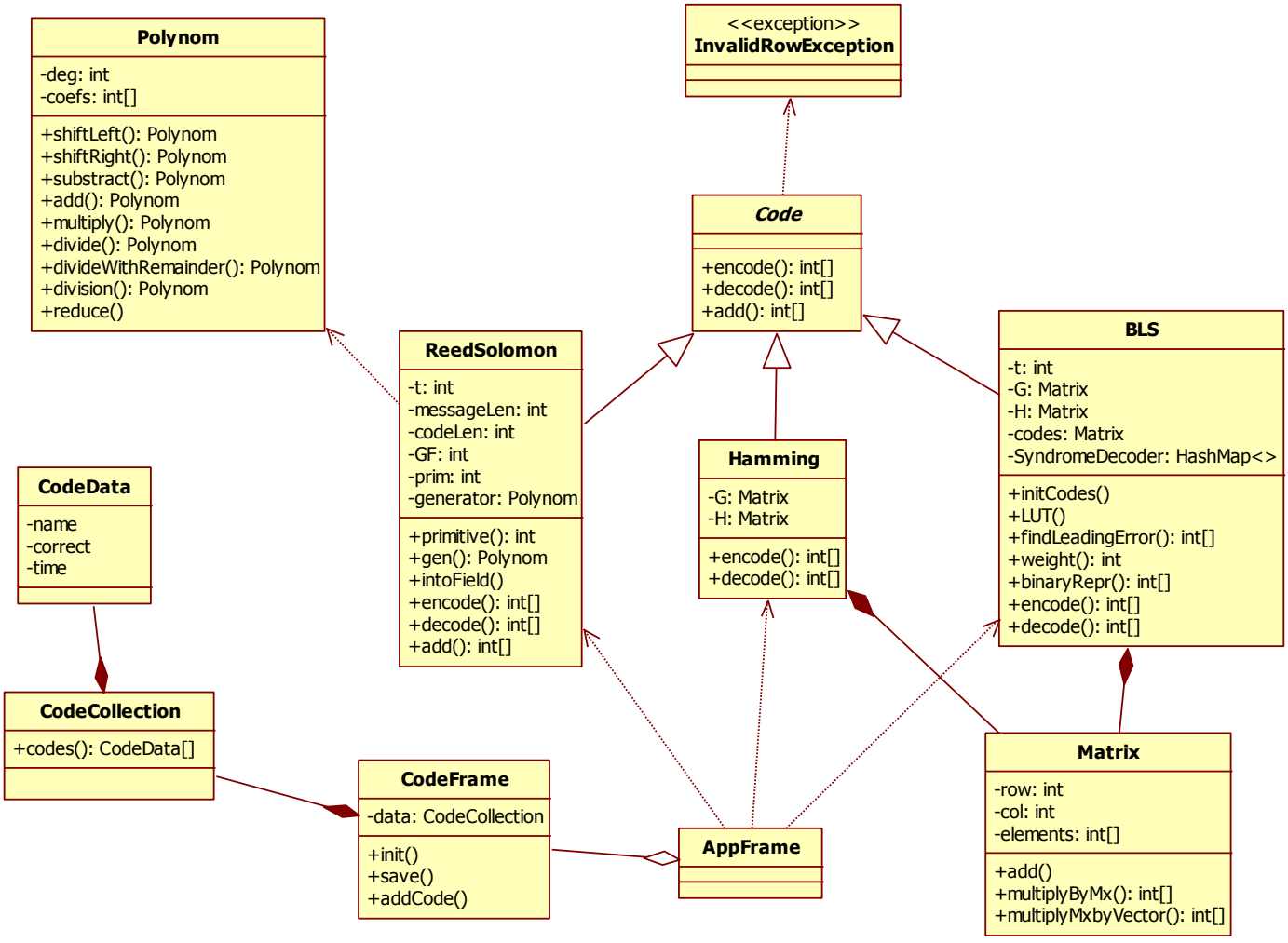
Hibajavító algoritmusok modell dokumentációja

A program három hibajavító algoritmust valósít meg: egy szisztematikus kódot használó bináris lineáris kódot (BLS), egy Hamming kódot és egy Reed-Solomon kódot. Megadhatjuk, hogy hány hiba javítására szeretnénk algoritmust az üzenet és hibavektorokkal együtt. Ezt követően látjuk az algoritmusok futásának végeredményét, melyeket a program automatikusan elment.

A BLS kódok, tekintve, hogy a legvalószínűbb kódszót választják ki egy hibacsoportból, t=1 esetén mindig helyes eredményt adnak, t>1 esetén pedig ritkán találkozunk hibátlan megoldással. Mivel nem találtam algoritmus a generátormátrix számolására, kézzel tettem meg. Így viszont nagy méretük miatt, nagyon lassúvá tették a programot. Végül olyan mátrixokat adtam meg melyek ugyan képesek a kijelölt hibák feloldására bizonyos esetben, a legtöbbször ez nem sikerül.
A Hamming kód minden esetben csak egy hibát tud javítani.
Az RS kódok dekódolásáért felelős hibacsapda algoritmus (ETA) csak akkor vezet megfelelő eredményre, ha a hibák az első t+1-es szegmensben tartózkodnak. Az algoritmus csak akkor tér vissza kivétellel, amennyiben végtelen ciklusba keveredne a hibák elrendeződéséből kifolyólag.

Az algoritmusokat egy-egy osztályban valósítottam meg, melyek mind egy absztrakt kódoló osztályból származnak (Code).

Az osztálydiagram:



Csontó Benjámin
JTB4Y1

Code osztály

encode	A származtatott osztályokban, itt valósul meg a kódoló algoritmus.
decode	A származtatott osztályokban itt valósul meg a dekódoló algoritmus.
add	Összead két egész számokat tároló ArrayList-et.

ReedSolomon osztály

Attribútumok:

t	Javítandó hibák száma.
messageLen	Üzenet hossza.
codeLen	Kódszóhossz.
GF	Az adott Galois-Test, melyben a műveleteket végezzük.
prim	A GF-hez tartozó primitív elem
generator	Generátorpolinom, mely az üzenet kódolásához szükséges

Metódusok:

primitive()	Meghatároz egy primitív elemet a GF-ben.
gen()	Kiszámolja a generátorpolinomot.
intoField()	A paraméterként kapott polinom együtthatóit leképezi az adott Galois-testre.
inGF()	Modulo GF. (Mivel a java-ban a „%” operátor maradékos osztást végez, így nem váltja át a negatív számokat.)
encode()	Lekódolja a kapott üzenetet a generátorpolinom segítségével.
decode()	Meghatározza a kapott vektorhoz tartozó üzenetet a hibacsapda algoritmus segítségével.
add()	Összead két polinomot GF-en belül.

BLS osztály

Attribútumok

t	Javítandó hibák száma.
G	Generátormátrix, az üzenet kódolásához.
H	Paritásellenőrző mátrix. A dekódoláshoz szükséges.
codes	G-hez tartozó kódszavak gyűjteménye.
SyndromeDecoder	Szindróma dekódolási táblázat, mellyel megtalálhatjuk a legvalószínűbb hibát.

Metódusok:

initCodes()	G alapján a kódszavak inicializálása.
LUT()	Szindrómadekódolási táblázatot hozza létre.
findLeadingError()	Egy hiba alapján létrehozza a hozzátartozó hibacsoportot és a legkisebb súlyút adja vissza.
weight()	Megadja egy vektor súlyát.
binaryRepr()	Átvált egy egész számot kettes számrendszerbe.

encode()	A generátormátrix segítségével lekódolja a kapott üzenetet.
decode()	A paritásellenőrző mátrix és a SyndromeDecoder táblázat segítségével meghatározza a legvalószínűbb üzenetet.

Hamming osztály

Attribútumok:

G	Generátormátrix, az üzenet kódolásához.
H	Paritásellenőrző mátrix. A dekódoláshoz szükséges.

Metódusok:

encode()	A generátormátrix segítségével lekódolja a kapott üzenetet.
decode()	A paritásellenőrző mátrix segítségével meghatározza az üzenetet.

Matrix osztály

A BLS és a Hammig kódokhoz szükséges mátrixműveleteket valósítja meg.

Attribútumok:

row	Ennyi sor van a mátrixban. (Tehát ilyen hosszú egy oszlop.)
col	Ennyi oszlop van a mátrixban.(Tehát ilyen hosszú egy sor.)
elements	Mátrix elemeit tárolja.

Metódusok:

add()	Hozzáad egy új sort a mátrixhoz. Amennyiben a sor hossza eltér a mátrix sorainak hosszától, kivételt dob.
multiplyByMx()	Megszorozza a kapott vektort a mátrixszal.
multiplyMxbyVector()	Megszorozza a kapott vektorral a mátrixot.

Polynom osztály

Az RS kódokhoz szükséges polinom műveleteket valósítja meg.
(Megjegyzés: Helyesen Polynomial, de mikor erre rájöttem már túl sok helyen kellett volna javítani.)

Attribútumok:

deg	Polinom fokszáma.
coefs	Együtthatókat tárolja fokszám szerinti növekvő sorrendben.

Metódusok:

shiftLeft()	Balra eltolja a coefs tagjait egyel, a legelsőt az utolsó helyére teszi.
shiftRight()	Jobbra eltolja a coefs tagjait egyel, a legutolsót az első helyére teszi.
substract()	Kiszámolja két polinom különbségét.
add()	Összead két polinomot.
multiply()	Összeszoroz két polinomot.

division()	Kiszámolja két polinom osztásának a hányadosát és a maradékát.
divide()	Visszaadja a hányadost.
divideWithRemainder()	Visszaadja a maradékot.
reduce()	Törli a legmagasabb értéken álló 0 együtthatókat, amíg első fogú nem lesz a polinom.

CodeData osztály

Algoritmusok futását tároló osztály.

CodeCollection osztály

Implementálja az AbstractTableModel interfészt. Ez alapján készül a futásokat tartalmazó táblázat.

CodeFrame osztály

Az algoritmusok futását táblázatosan jeleníti meg.

AppFrame osztály

GUI megvalósítása. Innen indul a program és itt hívódnak meg a kódoló osztályok.

InvalidRowException osztály

Amennyiben az üzenet hossza nem egyezik a kért paraméterrel, ezt a kivételt dobjuk.

A teszteseteket osztályonként valósítottam meg. Mivel a BLS kód t>1 esetén nem biztosít helyes kimenetelt, így formálisan azt nem teszteltem.