

Објектно оријентисано програмирање, 19.09.2016.

Напомена: Направити на Desktop-у директоријум са именом облика **OOP_InicijaliAsistenta_Prezime_Ime** и изабрати га за *workspace* директоријум при покретању Eclipse. **Пројекат и пакет назвати исто тако!!!**
Обавезно сачекати да неко од дежурних асистената прекопира Ваш рад!
Назначен је део који треба урадити да би се освојио праг. Такође, код **не сме** имати синтаксних грешака.
Време за рад: **2,5 сата**.

Задатак (Откривање грешке у кодираним бинарним порукама).

(**кратак опис задатка**) Кодирани бинарне поруке су подразумевано дужине **12 битова** и задате су у општем облику: **m8...m1c4...c1** (где су **m_i**, **i=8,...,1** битови поруке, а **c_i**, **i=4,...,1** контролни (кодирани) битови). Поступак описан у другом делу задатка може да открије **само један** погрешан бит у кодираној поруци. Порука је **неисправна** само ако је бит поруке погрешан. У свим другим случајевима је **исправна** (па и у случају када је погрешан један контролни бит).

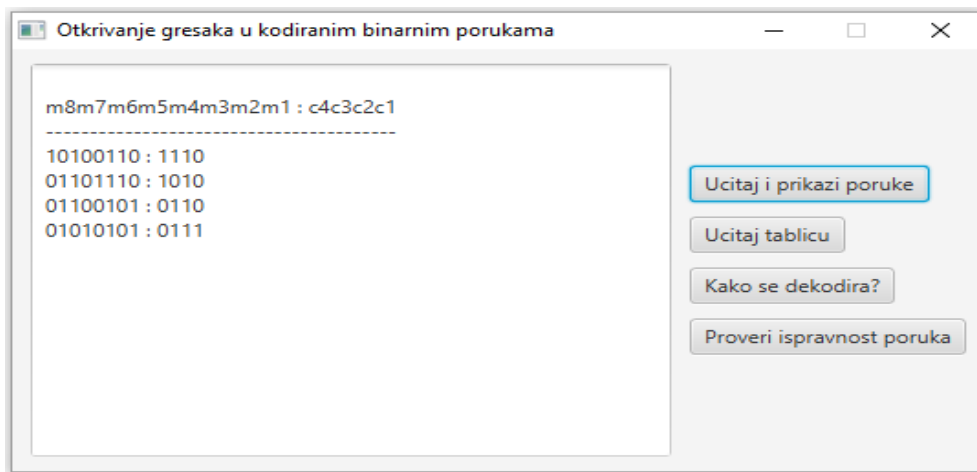
(**део за праг**) Написати апликацију чији графички кориснички интерфејс треба да изгледа као на слици 1.

НИЈЕ ДОЗВОЉЕНА УПОТРЕБА SCENEBuilder-A! Ономогућити ручни унос текста у текст-област. Притиском на дугме „**Учитај и прикажи поруке**“ треба да се омогући учитавање података из улазне датотеке *poruke.txt* (пример садржаја датотеке наведен је поред слике 1), а затим и приказивање података у текст области у формату датом на слици 1.

Сваки ред датотеке садржи један стринг од тачно 12 карактера који представља кодирану бинарну поруку. Првих осам карактера су битови поруке дати у поретку: **m8...m1**, а преостала четири су контролни битови дати у поретку: **c4...c1**.

За сваки прочитани ред улазне датотеке издвојене битове поруке и контролне битове приказати у облику датом на слици 1. Сматрати да је датотека у исправном формату.

Уколико датотека не садржи никакав садржај, у левој текст-области приказати поруку „**Нема порука у датотеци!**“.



```
101001101110
011011101010
011001010110
010101010111
```

Пример датотеке
poruke.txt

Слика 1

Направити помоћну класу **КодиранаПорука** са две чланице типа *String*: *bitoviM* (битови поруке) и *bitoviC* (контролни битови). Обезбедити одговарајући конструктор који прима све потребне податке, *get()* и *set()* методе и метод *toString()*, који враћа *String*-репрезентацију кодираних порука у формату **bitoviM : bitoviC**.

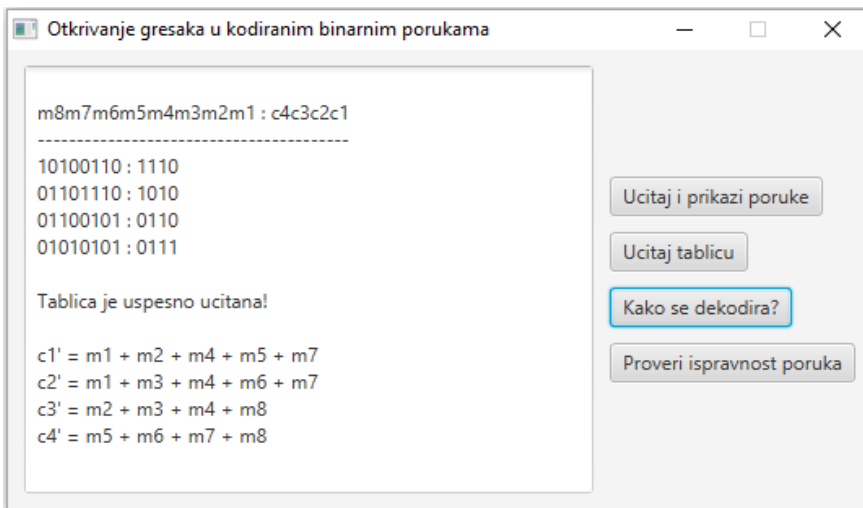
Обезбедити да се притиском на дугме „**Учитај и прикажи поруке**“, поред дела за праг, за сваку линију улазне датотеке прави објекат класе **КодиранаПорука** на основу прочитаних података и да се направљени објекти чувају у листи.

Притиском на дугме „**Учитај таблицу**“ омогућити учитавање података из улазне датотеке *tablica.txt* (садржај датотеке је дат поред слике 2) и исписати поруку „**Таблица је успешно учитана!**“ у текст-области. Датотека садржи таблицу која описује поступак за откривање грешке. Таблица има тачно 12 редова, а позиције редова броје се опадајуће од 12 до 1. На тих 12 позиција су распоређени битови поруке и контролни битови тако да се међусобно преплићу (у редоследу **m8m7m6m5c4m4m3m2c3m1c2c1**). Сваки ред садржи два *string*-податка: *бинарну четворобитну вредност позиције реда* и *ознаку бита* поруке (**m_i**, **i=8,...,1**), односно контролног бита (**c_i**, **i=4,...,1**) који је распоређен у датом реду.

Направити и попунити две мапе типа **Map<String, Integer>**, са подацима о бинарној вредности позиције у таблици и њој придруженом биту (за прву мапу биту поруке, а за другу контролном биту). Битове поруке и контролне битове представити њиховим редним бројевима (од 1 до 8 за битове поруке, тј. од 1 до 4 за контролне битове).

Притиском на дугме „**Како се декодира?**“ омогућити да се из улазне датотеке *formule.txt* прочитају и у текст-области прикажу формуле за декодирање порука, тј. за рачунање вредности **нових контролних битова** датих у поретку **c4'...c1'** (слика 2). Сваки ред датотеке садржи формулу за рачунање једног контролног бита **ci'** (**i=1,...,4**): прво се израчуна збир вредности свих оних битова **m** наведених иза знака '=', а потом и остатак при дељењу добијеног збира са 2. Сматрати да је датотека у исправном формату. Садржај је дат поред слике 2 (знак '+' се као делимитер наводи у облику '\\+'). Прочитане формуле за рачунање нових контролних битова сачувати у мапи **Map<Integer, List<Integer>>**, тако што се за сваки **с-бит** (тј. његов *редни број*) одговарајућа комбинација *редних бројева m-битова* чува у листи.

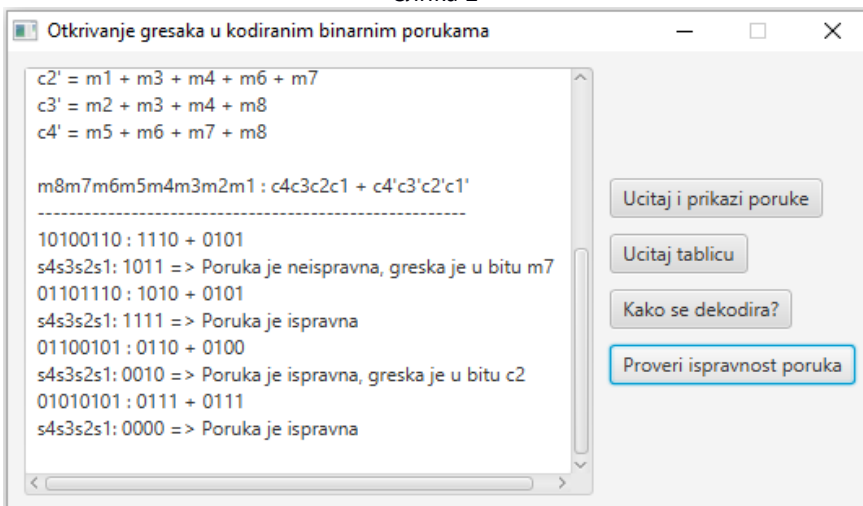
Притиском на дугме „**Провери исправност порука**“ омогућити проверу исправности сачуваних порука из листе. Прво треба одредити битове **c1',...,c4'** по горе описаном поступку, користећи податке из мапе **Map<Integer, List<Integer>>**. Добијене вредности сачувати као стринг облика **c4'c3'c2'c1'** и приказати га заједно са битовима поруке и постојећим контролним битовима у облику: **m8m7m6m5m4m3m2m1 : c4c3c2c1 + c4'c3'c2'c1'** (слика 3). Потом извршити проверу исправности. На основу стрингова **c4c3c2c1** и **c4'c3'c2'c1'** одређује се нови стринг **s4s3s2s1**, по формули: **si = (ci + ci') % 2, i=4,...,1**. Садржај стринга **s4s3s2s1** такође приказати у текст-области (слика 3). Уколико стринг **s4s3s2s1** одговара неком кључу у мапи **Map<String, Integer>** са **m-битовима**, порука је **неисправна** и грешка је у **m-биту** придруженом кључу. Уколико стринг **s4s3s2s1** одговара неком кључу у мапи **Map<String, Integer>** са **с-битовима**, порука је **исправна**, а грешка је у **с-биту** придруженом кључу. Иначе нема грешке и порука је **исправна**. У сва три случаја приказати извештај о исправности у текст-области према примеру на слици 3.



Слика 2

| | | |
|------|----|------------------------------|
| 1100 | m8 | c1' = m1 + m2 + m4 + m5 + m7 |
| 1011 | m7 | c2' = m1 + m3 + m4 + m6 + m7 |
| 1010 | m6 | c3' = m2 + m3 + m4 + m8 |
| 1001 | m5 | c4' = m5 + m6 + m7 + m8 |
| 1000 | c4 | |
| 0111 | m4 | |
| 0110 | m3 | |
| 0101 | m2 | |
| 0100 | c3 | |
| 0011 | m1 | |
| 0010 | c2 | |
| 0001 | c1 | |

tablica.txt formule.txt



Слика 3