**Објектно оријентисано програмирање, јун II, 13.07.2014.**

**Група А**

**Важно:На Desktop-у направити директоријум са именом облика <ВашеПрезиме><ВашеИме> и њега изабрати за workspace при покретању Eclipse. Пројекат назвати исто тако. Не остављати синтаксне грешке у коду (проблематичне делове кода ставити под коментаре).**

***Задатак (Велики цели бројеви).***

|  |
| --- |
| (***Део за праг***) Написати апликацију чији графички кориснички интерфејс треба да изгледа као на слици 1. Клик на дугме „**Učitaj brojeve**“ треба да омогући избор улазне датотеке (искључиво у .txt формату!) са локалног хард диска из којег се учитавају подаци задати у форми приказаној на следећој страни. Садржај датотеке чине произвољно велики (позитивни или негативни) цели бројеви. Сваки број је записан у посебном реду датотеке. Сматрати да је датотека у исправном формату. Иницијално треба да буде приказан садржај текућег директоријума корисника (тј. директоријума JavaFX пројекта)! Након избора документа: уколико је документ празан у текст области се исписује порука „**Nema brojeva!**“ (слика 2), а уколико документ садржи велике целе бројеве, онда се они исписују у текст области у формату приказаном на слици 3.  Обезбедити да се приликом клика на дугме „**Učitaj brojeve**“, парсирају линије датотеке, и да се у вектору чувају подаци о великим целим бројевима. За представљање великих целих бројева написати засебну класу **VelikiCeoBroj** која као податке чланове садржи знак броја и вектор његових цифара. Написати и одговарајуће конструкторе и метод **toString()** за приказ великог целог броја. Вектор великих целих бројева исписати на стандардни излаз у формату у ком је исписан у текст области на слици 3. |

|  |  |  |
| --- | --- | --- |
| Слика 1 | Слика 2 | Слика 3 |

У оквиру класе **VelikiCeoBroj**, поред потребних конструктора и методе **toString()**, написати следеће методе:

* методе неопходне за успешно сортирање колекције великих целих бројева
* метод:

**public static VelikiCeoBroj MnozenjeJednocifrenimBrojem(VelikiCeoBroj vcb, int broj);**

који као резултат враћа велики цео број који је добијен множењем броја **vcb** једноцифреним бројем **broj**.

Обезбедити да се приликом притиска на дугме „**Sortiraj**“ у текст области испишу сви бројеви сортирани у расућем поретку, у облику приказаном на слици 4.

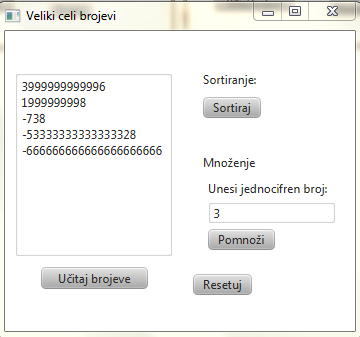
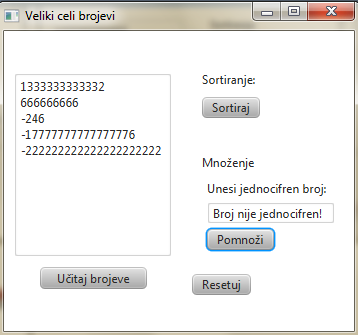
Обезбедити да се приликом притиска на дугме „**Pomnoži**“ у текст области испишу сви велики бројеви помножени једноцифреним бројем који се задаје у текст пољу (видети слику 5). Уколико се у текст поље унесе број који није једноцифрен, треба исписати поруку “Broj nije jednocifren!” (видети слике 6 и 7).

Уколико има више измена над вектором бројева (на пример, сортирање за којим следе два множења), свака следећа измена треба да буде изведена над већ измењеним вектором бројева. Дакле, ако се вектор прво сортира па множи, множење ће бити изведено над сортираним вектором. Приликом следећег множења, множење ће бити изведено над резултатом претходног множења (погледати слике 5-8).

Притиском на дугме „**Resetuj**“ обезбедити да се сви бројеви избришу из текст области.

**Важна напомена:** Делови задатка који се односе на сортирање, множење и ресетовање су независни и могу се решавати произвољним редоследом. Сваки део носи одговарајући број поена.

|  |  |  |
| --- | --- | --- |
| Слика 4 | Слика 5 | Слика 6 |



**Слика 7 Слика 8**

Пример садржаја улазног документа:

123

8888888888888888

-666666666666

111111111111111111111

-333333333