

Развој софтвера – јануар 2, 2019

Број индекса	Име и презиме

Испит траје 2 сата и 30 минута.

ПИСАТИ ЧИТКО - НЕЧИТКИ ЗАДАЦИ НЕЋЕ БИТИ ПРЕГЛЕДАНИ

Задатак	1	2	3	4	5	6	7	8	9	10	11	12	13	Укупно
Макс.	20	20	20	5	5	5	5	5	5	5	5	5	5	110

1. Испитати шта ради функција `opadajuci` (дата у фајлу `opadajuci.h`) и написати (бар) десет захтева (**REQUIRE** команди библиотеке **Catch**) у фајлу `test.cpp` који тестирају уобичајене и специјалне случајеве функције `opadajuci` (у фајлу је већ наведен тест за један случај).

Затим рефакторисати и оптимизаовати функцију `opadajuci` тако да више не користи **for** и **while** петље (није дозвољено користити ни **for**, ни **while**, ни **std::for_each**, ни рекурзију, ни **GOTO**).

2. Дат је костур функцијског објекта `transformator` који служи за конверзију слова у стрингу у мала или велика. Конструктор добија карактер **'m'** ако сва слова у стрингу треба претворити у мала, а карактер **'V'** ако сва слова у стрингу треба претворити у велика слова.

Написати оператор позива функцијског објекта (*call operator*) који добија један стринг као аргумент и враћа трансформисан стринг као резултат. Уколико се функцијском објекту више пута заредом проследи иста вредност, не трансформисати стринг поново, него чувати претходни резултат. У чланској променљивој `m_original` чувати последњи добијени аргумент, а у чланској променљивој `m_transformisan` чувати последњи израчунат резултат.

Написати оператор доделе који гарантује да ако се током његовог извршавања појави изузетак, да се ниједна чланска променљива у инстанци класе `transformator` којој се додељује нова вредност неје променила (инстанца на коју показује `this`), а да ће се изузетак пропагирати ономе ко је позвао оператор доделе.

Заменили употребу карактера **'m'** и **'V'** енумератором.

3. Написати **Qt** апликацију која представља део генерисања слова из игре погађања речи из игре Слагалица. У директоријуму `zad3` налази се **GIF** анимација која илуструје како апликација треба да се понаша, а на слици 1 приказ како треба да изгледа графички кориснички интерфејс.

На клик дугмета **Generisi** покрећу се 4 нити које врше генерисање насумичног слова 10 до 40 пута (укључујући оба). Након што нит генерише слово, емитује сигнал `letterGenerated(int, char)` којим обавештава које је генерисано слово, а на графичком интерфејсу се ажурира одговарајућа лабела. Нит потом спава 400 милисекунди. Након што нит заврши комплетан посао, емитује сигнал `generationComplete()` којим обавештава да је завршила свој посао.

Када се покрену нити, потребно је онемоућити да дугме **Generisi** дозволи даље кликање док све нити не заврше свој рад. Након што све нити заврше свој рад, потребно је поново омогућити да дугме **Generisi** прихвата кликове.

На клик дугмета **Izadji** излази се из апликације.

4. Објаснити намену градивних образаца за пројектовање.

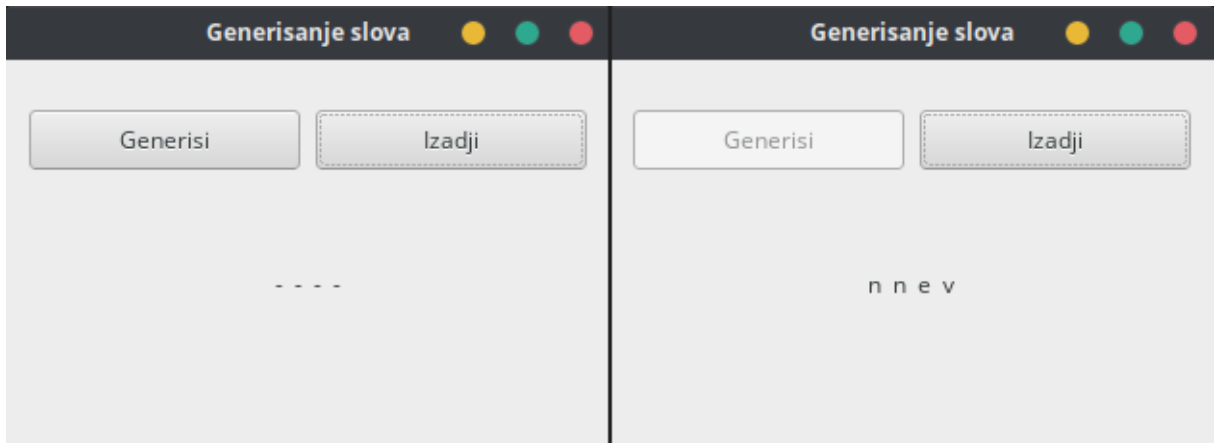


Рис. 1: Пример графичког интерфејса за задатак 3

5. Објаснити метод екстремног програмирања *"Игра планирања"*.
6. Који су основни мотиви за рефакторисање кода?
7. Објаснити правило дебаговања „Затражити туђе мишљење“.
8. Шта је дизајн софтвера? Шта је архитектура софтвера? Објаснити сличности и разлике.
9. Објаснити спрегнутост преко маркера.
10. Навести и објаснити слојеве тока догађаја код архитектура заснованих на догађајима.
11. Објаснити детаљно операцију чекања нити. Како се имплементира помоћу библиотеке *Qt*?
12. Објаснити функцију `async` и шаблон `future` стандардне библиотеке C++-а.
13. Објаснити метрику *стабилност пакета*.