

Трећа као лаб. вежба из Објектно оријентисаног програмирања 1

1) Написати на језику C++ следеће класе (класе опремити оним конструкторима, деструктором и операторима доделе који су потребни за безбедно и ефикасно коришћење класа):

- **Скуп** се састоји од произвољног броја знакова (`char`). Ствара се са задатом C++ ниском (`string`), чији се знаци одмах додају. Могуће је додати знак у скуп (`skup+=znak`), при чему се додавање знака који се већ налази у скупу игнорише, као и испитати да ли се знак налази у скупу (`skup(znak)`). Скуп није могуће копирати ни на који начин.
- **Реч** се ствара са задатом C++ ниском (`string`) и састоји се само од слова енглеског алфавета. Могуће је одредити дужину речи (`+rec`), као и број слогова речи (`~rec`). Број слогова речи једнак је броју носилаца слогова речи. Носилац слога је сваки самогласник у скупу {a, e, i, o, u} или сваки сонант у скупу {l, n, r}, ако се ни испред ни иза њега не налази самогласник, а за слова l и n, ни ако се иза њих не налази слово j. Могуће је одредити позицију слова n-тог носиоца слога у речи (`rec(n)`), при чему позиције слова крећу од 0. Вредност `n==0` одређује први слог на почетку речи, позитивне вредности n означавају претрагу од почетка речи, а негативне вредности n од краја речи. У случају неуспешне претраге повратна вредност је -1. Две задате речи се римују (`rec1^rec2`) ако им се поклапају одговарајућа слова последња два слога, односно последњег слога ако је нека од речи једносложна. Провера поклапања слова врши се од носилаца претпоследњих слогова у речима, односно носилаца последњих слогова, ако је нека од речи једносложна. Реч је могуће учитати из једне линије улазног тока (`it>>rec`), при чему се игноришу сви знаци који нису слова енглеског алфавета и при чему се губи претходни садржај речи. Реч се исписује у излазни ток (`it<<rec`) исписивањем њених слова.
- **Стих** је секвенца речи произвољне дужине. Ствара се празан, а након тога је могуће додавати појединачне речи на крај стиха (`stih+=rec`). Могуће је одредити дужину стиха изражену у броју његових речи (`+stih`), као и број слогова стиха изражен у збиру слогова његових речи (`~stih`). Могуће је дохватити реч са задате позиције (`stih[ind]`), обрисати реч са задате позиције (`stih(ind)`) и уметнути реч испред речи на задатој позицији (`stih(rec, ind)`). Два задата стиха се римују (`stih1^stih2`) ако им се римују последње речи. Стих је могуће учитати из једне линије улазног тока (`it>>stih`) читањем речи из те линије. Стих се исписује у излазни ток (`it<<stih`) исписивањем његових речи одвојених тачно једним знаком размака, у једном реду.
- **Строфа** је секвенца стихова ограниченог капацитета (броја стихова), који се задаје при стварању и који може да се дохвати. Ствара се празна, а након тога је могуће додавати појединачне стихове на крај строфе (`strofa+=stih`), при чему је операција без ефекта у случају да се покуша додавање стиха у пуну строфу или стиха са различитим бројем слогова од стихова који се већ налазе у строфи. Приликом додавања стиха у строфу прави се копија стиха који се додаје. Могуће је уклонити последње додати стих (`-strofa`), при чему је уклањање из празне строфе операција без ефекта, као и одредити дужину строфе изражену у броју њених стихова (`+strofa`). Могуће је дохватити стих са задате позиције (`strofa[ind]`), као и заменити места стиховима са задатих позиција (`strofa(i, j)`). Могуће је одредити да ли се стихови строфе римују (`*strofa`), као и једнословну ознаку врсте строфе. Строфа се исписује у излазни ток (`it<<strofa`) исписивањем њених стихова у појединачним редовима. Може се проверити да ли је строфа ритмична (`!strofa`). Строфа је ритмична уколико су сви стихови у њој исте дужине (имају исти број речи) и уколико речи на респективним позицијама у сваком стиху имају исти број слогова и уколико се строфа римује.
- **Катрен** је строфа капацитета 4. Стихови катрена се римују, ако је број стихова једнак капацитету и ако се први стих римује са последњим, а други са претпоследњим. Врста катрена је К.

- **Терцина** је строфа капацитета три. Стихови терцине се римују ако је број стихова једнак капацитету и ако се прва два стиха римују. Врста терцине је **Т**.
- **Песма** је секвенца строфа одређене дужине која је ствара називом и именом писца. Сви подаци могу да се дохвате. Може се додати строфа у песму (`pesma+=strofa`). Грешка је уколико се покуша додавање строфа које не испуњавају услове. Може се проверити ритмичност песме (`!pesma`). Песма је ритмична ако су јој све строфе ритмичне. Не може се копирати ни на који начин. Песма се на излазни ток (`it<<pesma`) исписује исписом назива песме и писца а затим строфа у новом реду.
- **Сонет** је песма од 14 стихова. Сонет садржи два катрена и две терцине, тим редом. Грешка је уколико се покуша додавање погрешним редоследом, додавање строфа које нису катрен/терцина или терцина чији се последњи стихови не римују.
- **Збирка** песама садржи произвољан број песама. Може се додати песма у збирку. Грешка је ако се покуша додавање песама различитих писаца у збирку. Може се дохватити број песама у збирци. Може се дохватити одређена песма из збирке (`zbirka[pesma]` или `zbirka[redni_broj]`). Уколико песма не постоји у збирци или је број ван опсега, операција је без ефекта. Збирка се исписује на излазни ток (`it<<zbirka`) исписивањем песама у новим редовима.

Написати главну функцију која учита неколико стихова са стандардног улаза, дода их у катрен и ~~непинне га на стандардном излазу~~, затим креира још један катрен и две терцине, додаје у их у два различита сонета. Од тих сонета је потребно направити збирку и исписати је, а затим дохватити други сонет и њега засебно исписати. За тестирање се може користити сонет [Слушање Јована Дучића](#).

НАПОМЕНЕ:

- Предвиђено време израде је 135 минута.
- Није дозвољено коришћење литературе.