Задачи за задължителна самоподготовка по Обектно-ориентирано програмиране

Конструктори

email: kalin@fmi.uni-sofia.bg

25 март 2016 г.

1. Да се дефинира клас `Rat`, описващ рационално число. За класа да се дефинират оператори за събиране и умножение на рационални числа, както и подходящи конструктори. Да се дефинира функция

   ```
   Rat poly (Rat coef[], int n, Rat x)
   ```

   където `coef` е масив с `n + 1` рационални коефициента $a_0, a_1, ..., a_{n-1}, a_n$, а `x` е рационално число.

   Функцията да намира стойността на полинома $P(x) = a_0x^n + a_1x^{n-1} + ...a_{n-1}x + a_n$.

   Да се реализира и изпълни подходящ тест.

2. Да се дефинира клас Word, описващ дума, съставена от не повече от 20 символа от тип char. Класът да съдържа следните операции:

   - оператор `[]` за намиране на `i`-тия пореден символ в думата
   - оператори `+` и `+=` за добавяне на един символ в края на думата. Ако думата вече има 20 символа, операторите да нямат ефект
   - оператори `<` и `==` за сравнение на думи спрямо лексикографската наредба

- подходящи конструктори

Да се реализира и изпълни подходящ тест за класа и неговите методи.

3. Да се реализира клас `NumbersSummator`, който поддържа сума на цели числа. При създаване на обект от класа, съответната му сума да се инициализира с число, което се подава като аргумент на конструктора. За класа да се реализират следните методи:

   - sum, който връща текущата стойност на сумата
   - add, увеличаващ сумата с дадено число
   - sub, намаляващ сумата с дадено число
   - num, връща колко пъти сумата е била променяна
   - average, връщащ средното аритметично на всички числа, с които сумата е била променяна.

   *Забележка:*Функционалност извън тези 4 метода, като например съхраняване на отделните числа от поредицата, не е необходима. Пример:

```
NumbersSummator seq1 (10);
seq1.add (10);
seq1.add (5);
seq1.sub (15);
cout << seq1.sum() ; //->10 (10+10+5-15)
cout << seq1.average(); //->0 (10+5-15)/3
```

4. Да се дефинира клас `BrowserHistory`, който съдържа информация за историята на посещението до най-много `N` Web сайта. `N` е параметър на конструктора на класа. За целта да се реализира структура `HistoryEntry`, описваща едно посещение на сайт чрез:

   (а) Месец от годината, през който е посетен сайтът;

   (б) Неговото URL.

   Класът codeBrowserHistory да поддържа следните операции:

   - Метод за добавяне на нов сайт към историята. Информацията за всеки сайт се въвежда от клавиатурата

- Оператор += с параметър `HistoryEntry`, добавящ сайт към историята
- Метод за отпечатване на информацията за всички сайтове в историята
- Метод, който по даден месец от годината намира броя на сайтовете, посетени през този месец
- Намиране на този месец от годината, в който има най-много посетени сайтове
- Премахване на най-скоро добавеният сайт в историята
- Оператор +, който обединява двете истории

Да се реализира и изпълни подходящ тест за класа и неговите методи.