

Консултация за второ контролно по ООП

Задача 1:

Крайна редица.

а)[15 т.] Да се реализира шаблон на клас Sequence, представящ крайна редица от елементи от даден тип T, дефинирана чрез:

1. Първи елемент на редицата a_0
2. Брой елементи k
3. Функция $\text{next}: T \rightarrow T$, генерираща поредните елементи. Т.е. $a_{i+1} = \text{next}(a_i)$ за $0 \leq i < k-1$.

Например, редицата `Sequence<int> even10(0,10,p2)`, където $p2(x)=x+2$, се състои от първите 10 неотрицателни четни числа 0, 2, ..., 18.

Предефинирайте оператора за индексване (`[]`) на Sequence.

б)[25 т.] За шаблона Sequence да се реализира прост forward интератор само за четене, така че следния цикъл да извежда елементите на редицата:

```
for (int x : even10)
{ std::cout << x << " "; }
```

Задача 2: Даден проект се състои от задачи, които могат да бъдат прости и съставни.

а) [5 т.] Да се разработи клас SimpleTask за проста задача, която се описва от свободен текст и има време за изпълнение в дни.

б) [25 т.] Да се разработи клас CompositeTask, описващ съставна задача.

- Съставната задача има наименование

- Към съставната задача могат да се добавят подзадачи, които може да са както прости, така и съставни

Реализирайте допълнителни класове и методи, необходими за решението на задачата, по ваша преценка.

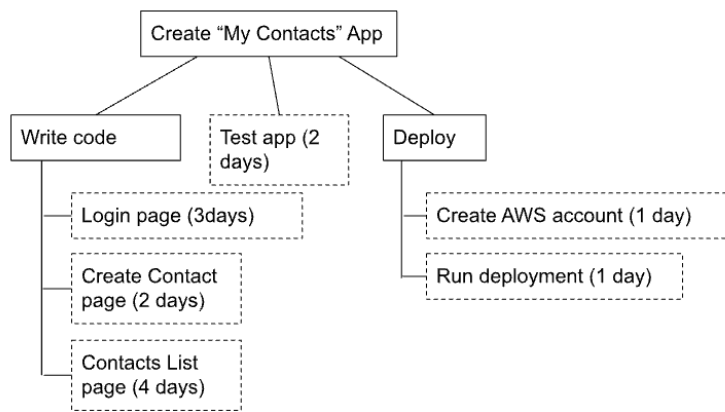
Да се реализират всички необходими конструктори, деструктори и оператори по ваша преценка.

Внимание: към решението на задачата се очаква реализация на виртуален конструктор за копиране (clone, deep copy) и неговото използване.

в) [5 т.] CompositeTask да предоставя метод totalTime, намиращ времето за изпълнение на всички подзадачи.

г) [15 т.] CompositeTask да предоставя метода printPlan, който извежда списъка от всички прости задачи, от които се състои задачата, включително вложените в нейни подзадачи, в реда, в който трябва да бъдат изпълнени. Реда на изпълнение на подзадачите на дадена съставна задача следва реда на добавянето им към съставната задача.

д) [10 т.] На диаграмата са дадени задачи и подзадачи. С непрекъснати линии са отбелязани съставните задачи, а с прекъснати – простите. Да се напише примерна main функция, построяваща задачата “Crate My Contacts App” и изпълняваща нейните методи totalTime и printPlan.



Изходът на printPlan за задачата Create “My Contacts” App е:

Login page

Create Contact page

Contacts List page

Test app

Create AWS account

Run deployment

Задача 3:

В софтуерна фирма има два вида служители – програмисти и мениджъри. Отдел “личен състав” поддържа следната информация за всеки от програмистите:

- име;
- стаж (в месеци);
- дали знае C++;
- дали знае C#

и следната информация за всеки от мениджърите:

- име;
- стаж (в месеци);
- колко човека управлява.

Да се напише програма, която позволява на отдел “личен състав” да поддържа списък с всички програмисти и мениджъри във фирмата. Програмата да може да извършва следните операции:

- постъпване на нов служител;
- напускане на служител;
- извеждане на списък с данни за всички служители.

Задача 4:

Да се реализира абстрактен базов клас Set, описващ множество от цели числа (int) с една единствена операция:

`bool member(int x)`

Операцията проверява дали дадено число е елемент на множеството. Да се реализират следните наследници на `Set`:

а) `EvenNumbers`: множеството на всички четни числа от тип `int`

б) `DynamicSet`: множество, което позволява добаяне на елементи чрез операция `addElement(int)`

в) `SetUnion`: обединение на произволен брой множества с операция `addSet([подходящ тип] s)`, където `s` може да е от всеки един от трите вида множества(*)

Да се реализират необходимите конструктори и деструктори, включително и виртуални такива.

(*)Множеството `s` може да се добави към обединението във вида, в който е при извършване на `addSet`. Т.е. бъдещи промени по `s` не е нужно да имат ефект върху обединение, което съдържа `s`.