



Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика

ПОПРАВИТЕЛЕН ИЗПИТ

курс Обектно-ориентирано програмиране

специалност Софтуерно инженерство

27.08.2018 г.

Времетраене: 3(4) часа

Изисквания за предаване:

- Предаване на решенията от изпита става като .zip архив със следното име:
Exam_SI_(курс)_(група)_(факултетен_номер), където:
 - (курс) е цяло число, отговарящо на курс (например 1);
 - (група) е цяло число, отговарящо на групата Ви (например 1);
 - (факултетен_номер) е цяло число, отговарящо на факултетния Ви номер (например 63666);
- Архивът да съдържа само изходен код (.cpp и .h файлове) с решение отговарящо на условията на задачите, като файловете изходен код за всяка задача трябва да са разположени в папка с име (номер_на_задача);
- Качване на архива на посоченото място в Moodle;

Пример за .zip архив за изпита: Exam_SI_1_1_63666.zip

Задача 0. (Задача за допускане) Въпроси и тестове

35 т.

- a. Да се реализира абстрактен базов клас `Question`, който описва следните операции върху въпрос от тест:
- `ask`, която извежда въпроса и въвежда негов отговор;
 - `grade`, която оценява въпрос и връща броя точки.

Да се реализират следните видове въпроси:

- b. `YesNoQuestion`, който описва въпрос с два възможни отговора: да или не. При конструиране се задават текст, точки и верен отговор. Въпросът дава 0 т. при грешен и пълен брой точки при верен отговор.
- c. `MultipleChoice`, който описва въпрос с няколко възможни отговори, от които някои са верни. При конструиране се задават текст, точки x , възможни и верни отговори. Всеки правилно посочен отговор добавя x точки, а всеки погрешно посочен отговор отнема x точки.
- d. `OpenQuestion`, който описва въпрос със свободен текст. При конструиране се задават текст и брой точки. При оценяване се извеждат въпроса и отговора и се въвежда оценка в проценти. Ако отговор не е даден, въпросът автоматично се оценява с 0 т.
- e. Да се реализира клас `Test`, представящ тест с въпроси от всякакъв вид. Да се реализират функциите:
- `addQuestion`, която добавя нов въпрос към теста;
 - `doTest`, която задава всички въпроси подред и събира отговорите им;
 - `gradeTest`, която оценява теста и връща броя събрани точки.
- f. Да се напише главна програма, в която:
- да се дефинират няколко въпроса (от всеки вид поне по 1);
 - да се дефинира тест и да се добавят въпросите към него;
 - да се изпълни теста;
 - да оцени направения тест.

Забележка: Разрешено е да ползвате класове от STL библиотеката.

Задача 1. Множества

35 т.

Да се напише шаблон на клас **Set** (множество), който представя множество от елементи от тип **T**. Да се реализират:

- Конструктор без параметри – създава празно множество от елементи от тип **T**;
- Метод **insert** – вмъква елемент в множеството. Ако елементът е вече в множеството – не прави нищо;
- Метод **erase** – изтрива елемент от множеството;
- Метод **contains** – проверява дали зададен елемент принадлежи на множеството и връща **true** или **false**;
- Метод **size** – връща броя на елементите в множеството;
- Оператор **[]** – връща елемент от множеството със съответен индекс;
- Метод **union** – връща обединението на 2 множества;
- Метод **intersection** – връща сечението на 2 множества;

Да се напише главна програма, която илюстрира работата с множества от цели числа и реални числа.

Забележка: Не е разрешено да ползвате класове от STL библиотеката. Да се осигури правилното функциониране на методите от “голямата четворка”.

Задача 2. Метеорологична станция и наблюдатели

35 т.

a) Да се реализира клас **MeteoStation** (Метеорологична станция). В метеорологичната станция се поддържат данни за времето за даден град и има следните член-данни:

- **city** – име на град (низ – например **Sofia** или **Varna**);
- **temperature** – текуща температура в града (реално число).

Да се реализират:

- конструктор;
 - функции за достъп – **getCity** и **getTemperature**;
 - мутатор **setTemperature** – променя текуща температура в града.
- b) Освен това има наблюдатели на времето за дадения град (вестници, телевизии, радиа и др.). Метеорологичната станция поддържа списък с указатели към наблюдателите. Да се реализират следните методи:
- **register** – добавя (регистрира) наблюдател към списъка (забележка: методът приема за аргумент указател към наблюдател и добавя указателя към списъка);
 - **unregister** – изтрива наблюдател от списъка;
 - **notify** – уведомява всички регистрирани наблюдатели за промяна на времето (температурата) в града, като извиква метод **update** на наблюдателя със съответни параметри. Методът **notify** трябва да се извиква при всяка промяна на температурата от **setTemperature**.

- c) Да се дефинира абстрактен клас **Observer** (наблюдател), който има чисто виртуален метод **update** за промяна на времето с параметри: име на град и температура.
- d) Да се дефинира производен клас **NewspaperObserver** (вестник – наблюдател, аналогично могат да се дефинират класове **TVObserver**, **RadioObserver**), който има следните член-данни:
- **name** - име на вестника (низ)
 - **city** – име на град;
 - **temperature** – текуща температура в града.

Да се реализират следните методи:

- конструктор
- метод **update**, който обновява данните за времето за града и извежда:

Update <name>: The temperature in <city> is <temperature>

- метод **printMeteo**, който извежда на екрана

Newspaper <name>: The temperature in <city> is <temperature>

- e) Да се реализира главна програма, която
- Създава обект от клас **MeteoStation**
 - Създава няколко обекта от клас **NewspaperObserver**
 - Регистрира наблюдателите в обекта на **MeteoStation**
 - Променя текущата температура в града
 - Извиква **printMeteo** на наблюдателите
 - Изтрива наблюдател от списъка с наблюдатели на обекта на **MeteoStation**
 - Променя текущата температура в града
 - Извиква **printMeteo** на наблюдателите

Забележка: Разрешено е да ползвате класове от STL библиотеката. Да се осигури правилното функциониране на методите от “голямата четворка”.