



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

ИЗПИТ

курс Обектно-ориентирано програмиране
специалност Софтуерно инженерство
летен семестър 2018/2019 г.

Сесия 2

Времетраене: 2 часа и 30 минути

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 1. Матрица (40т.)

Да се реализира шаблон на клас `Matrix`, който представлява двумерен масив от елементи.

Да се реализират следните методи:

- Конструктор - заделя матрица с размер 2×2 (всички елементи са със стойност по подразбиране)
- Конструктор с параметри - размер на матрицата (всички елементи са със стойност по подразбиране)
- Конструктор за копиране
- Оператор=
- Деструктор
- `setAt(unsigned int x, unsigned int y, T element)` - променя елемента на позиция (x,y) (x и y започват от **1**)
- `getAt(unsigned int x, unsigned int y)` - връща елемент на позиция (x,y) (x и y започват от **1**)
- `transpose()` - транспонира матрицата

Примерни стойности и вход/изход:

```
Matrix<int> test1(2,2);
test1.setAt(1,1,5);
test1.setAt(1,2,3);
test1.setAt(2,1,4);
test1.setAt(2,2,1);
/*
(5,3)
(4,1)
*/
test1.transpose();
/*
(5,4)
(3,1)
*/
std::cout << test1.getAt(1,2); //4
```

Забележка: Не е разрешено да ползвате класове от STL библиотеката.

Задача 2 Ресторант (40т.)

Да се реализира софтуер за ресторант. В ресторанта ще се предлагат следните типове артикули - **храна** и **напитка**

Да се напишат следните класове и техните член-данни/методи:

Клас `RestaurantItem` (абстрактен):

- Член-данни:
 - Тип на продукта (enum)
 - Наличност (цяло положително число) (0 по подразбиране)
 - Цена (дробно число) (0 по подразбиране)
- Методи:
 - `Get()` за всяка член-данна
 - `Set()` за всяка член-данна
 - Конструктор с параметри

Клас `Food`: (наследява `RestaurantItem`)

- Член-данни:
 - Тип на продукта: храна
 - Име на храната (string) (0 по подразбиране)
 - Грамаж (дробно число) (0 по подразбиране) (в кг)
- Методи:
 - `getName()` - връща името на храната
 - `Set()` за всяка член-данна
 - Конструктор с параметри
 - Оператор `>` - сравнява два плода по грамаж
 - `print()` - извежда информацията за продукта на конзолата

Клас `Drink`: (наследява `RestaurantItem`)

- Член-данни:
 - Тип на продукта: напитка
 - Име на напитката (string)
 - Алкохолно съдържание (в %) (0 по подразбиране)
- Методи:
 - `getName()` - връща името на напитката
 - `Set()` за всяка член-данна
 - Конструктор с параметри
 - Оператор `>` - сравнява два продукта по алкохолното
 - `print()` - извежда информацията за продукта на конзолата

Клас `Restaurant`:

Служи като контейнер за артикулите в ресторанта.

- Член-данни:
 - Хетерогенен контейнер за артикулите в магазина
- Методи:
 - Добавяне на продукт (приема продукта като аргумент)
 - Премахване на продукт (приема индекс)
 - Промяна на цена на продукт (приема индекс и цена)
 - Промяна на име на продукт (приема индекс и ново име)

- Принтира всички продукти
- Извежда всички безалкохолни напитки

Примерни стойности и вход/изход:

```
Restaurant restaurant;
```

```
Food t1;  
t1.setName("Pizza");  
t1.setSize(0.400);  
t1.setAmount(30);  
t1.setPrice(5.50);
```

```
Drink t2;  
t1.setName("Water");  
t1.setAlcohol(0);  
t1.setAmount(100);  
t1.setPrice(2.80);
```

```
restaurant.add(t1);  
restaurant.add(t2);  
restaurant.changePrice(1, 1.80);
```

Забележка: Разрешено е да ползвате класове от STL библиотеката.