



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

## ДОМАШНА РАБОТА №1

курс Обектно-ориентирано програмиране  
специалност Софтуерно инженерство  
летен семестър 2017/2018 г.

СРОК ЗА ПРЕДАВАНЕ: посочен в Moodle

Изисквания за предаване:

- Предаване на домашното в указания срок от всеки студент като .zip архив със следното име: (номер\_на\_домашно)\_SI\_(курс)\_(група)\_(факултетен\_номер), където:
  - (номер\_на\_домашно) е цяло число, отговарящо на номера на домашното за което е отнася решението (например 1);
  - (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), където (номер\_на\_задача) е номера на задачата към която се отнася решението;
- **Не е разрешено** да ползвате класове от библиотеката STL като std::string, std::vector, std::stack и др.
- Качване на архива на посоченото място в Moodle;

Пример за .zip архив за домашно: 1\_SI\_1\_1\_63666.zip

## Задача 1. Калкулатор за изрази в обратен полски запис

3 точки

Обратният полски запис е математически запис (нотация) на изрази, в която всеки оператор следва всички негови операнди.

Например стандартният запис на израза

$$5 + ((1 + 2) * 4) - 3$$

може да бъде записан по следния начин с обратен полски запис:

$$5\ 1\ 2\ +\ 4\ *\ +\ 3\ -$$

Реализирайте калкулатор на изрази над цели или реални числа, записани в обратен полски запис, като за целта:

- Реализирайте клас **Stack** (разширяващ се стек или свързан стек) от числа (цели или реални)
- Напишете функция **calculate**, която приема като аргумент низ (с дължина максимум 100 знака) – аритметичен израз над цели (или реални числа) записан в обратен полски запис с оператори са +, -, \*, / и % (само за цели числа). Функцията изчислява израза и връща като резултат число.
- Напишете главна програма за тестване на калкулатора.

## Задача 2. Browser History

3 точки

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:

- Дата (месец от годината), през който е посетен сайтът (за простота използвайте месец вместо дата);
- Неговото URL - низ, с произволна дължина

Класът **BrowserHistory** да поддържа следните операции:

- Метод за добавяне на нов сайт към историята. Информацията за всеки сайт се въвежда от клавиатурата.
- Метод с параметър **const HistoryEntry&**, добавящ сайт към историята. Ако браузерът е стигнал лимита си от към памет, само изписва грешка и не добавя нов сайт към хронологията си. Важи и за по-горната операция.
- 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.
- Метод, който конкатенира (обединява) два обекта от клас BrowserHistory и връща нов обект - НЕ променя предишните два.

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

Да се моделира онлайн магазин (**Store**) за определени продукти (тениски, ризи, маратонки, GSM, лаптопи, автомобилни гуми, и др.). За целта:

- Изберете продукт, за който ще реализирате магазина и определете характеристиките на продукта (например марка, модел, категория, цвят, размер, операционна система, памет, и др.)
- Да се дефинира клас **Product**, който има следните член-данни:
  - **SKU** (stock keeping unit) – уникален номер на продукта (задължително);
  - **Brand** – марка (задължително)
  - **Model** – модел (задължително)
  - **Category** – категория (ако е приложимо)
  - **Price** – единична цена (задължително)
  - **Count** – брой налични продукта в магазина (задължително)
  - Други член данни в зависимост от избрания продукт и определените характеристики.
  - Използвайте коментари за класа и член-данните, за да е ясно какъв е избраният продукт и какви са неговите характеристики.
- Да се дефинират съответни методи за създаване, четене, запис и промяна на продукти;
- Да се дефинира клас **Store**, който съдържа динамичен списък (масив) от продукти в магазина, заедно с техния брой. Да се дефинират съответни методи за:
  - Добавяне на нов продукт;
  - Изтриване на продукт;
  - Промяна на продукт;
  - Извеждане на екрана на списък с наличните продукти и техния брой;
  - ~~○ Записване на всички продукти във файл;~~
  - ~~○ Зареждане на продуктите от файл.~~
- Да се дефинира главна програма, реализираща следното меню:
  - A     Add new product
  - X     Delete product
  - C     Change product
  - D     Display products
  - ~~S     Save products~~
  - ~~L     Load products~~
  - Q     Quit