



# ООП – Практикум: Домашна работа №1

## Задача 1: Дати

Реализирайте клас за работа с дати. Класът трябва да има 3 член-данни:  
Ден, месец и година. Класът трябва да има:

1. Конструктор по подразбиране, който прави датата на 01.01.2000.
2. Конструктор, който приема като аргумент ден, месец и година.
3. Метод, който приема като параметър число  $n$  и увеличава датата с  $n$  дни ( $n \in \mathbb{N}$ ).
4. Метод, който приема като параметър число  $n$  и намалява датата с  $n$  дни ( $n \in \mathbb{N}$ ).
5. Метод, който връща истина, ако годината е високосна и лъжа, ако не е.
6. Метод, който връща колко дни остават до Коледа.
7. Метод, който връща колко дни остават до края на годината (до 01.01.XXXX, където XXXX е настоящата година увеличена с 1).
8. Метод, който приема като параметър дата (напр. рожден ден или друго събитие) и връща колко дни остават до него.
9. Метод, който приема като параметър дата и връща истина само ако тя е след текущата.

### Примери

```
{  
    Date d1(31, 3, 2014); // 31.03.2014  
  
    Date d2; //01.01.2000  
  
    d1.addDays(1); // 01.04.2014  
  
    d1.removeDays(1) //31.03.2014  
  
    d1.DaysToXmas //269  
  
    d1.isLeapYear(); //false (2014 is not a leap year)  
  
    d1.isLaterThen(d2); //false  
}
```





# ООП – Практикум: Домашна работа №1

## Задача 2: Ю-Ги-О!

Сето Кайба организира турнир по Ю-Ги-О!, но е съкратил бюджета в компютърния отдел и помолил нас да организираме базата данни с всички дуелисти в турнира.

Трябва да реализирате 2 класа Ю-Ги-О! карти:

- "Карта чудовище"
  - Име (низ с дължина до 25 символа)
  - Атакуващи точки (неотрицателно целочислено число)
  - Защитни точки (неотрицателно целочислено число)
- "Магическа карта"
  - Име (низ с дължина до 25 символа).
  - Ефект (низ с дължина до 100 символа).
  - Тип (може да е един следните 3: "trap", "buff", "spell")

Реализирайте клас тесте (deck). Тестето трябва да има:

1. Масив от карти чудовища (максимално – 20 карти).
2. Масив от магически карти (максимално – 20 карти).
3. Метод, който връща като резултат броя на магическите карти в тестето.
4. Метод, който връща като резултат броя на картите чудовища в тестето.
5. Метод, който добавя магическа карта към тестето.
6. Метод, който добавя карта чудовище към тестето.
7. Метод, който променя магическа карта от тестето, като за параметри приема индекс и нова карта.
8. Метод, който добавя карта чудовище към тестето, като за параметри приема индекс и нова карта.

Реализирайте клас дуелист (Duelist). Той трябва да има:

1. Име (Низ с произволна дължина).
2. Тесте(deck)
3. Дуелистът трябва да има същите методи като в deck, с които да променяме тестето му





# ООП – Практикум: Домашна работа №1

## Примери

---

```
{  
  
    //40 cards = 20 MonsterCards/20 MagicCards  
    Deck d;  
  
    // 4-th card in the Monster deck becomes "Blue-eyes white dragon" with  
    //ATK: 3000 DEF: 2500  
    d.changeMonsterCard(4, "Blue-eyes white dragon", 3000, 2500);  
  
    // 5-th card in the Magic deck is added/changed to "Monster reborn"  
    d.changeMagicCard(5, "Monster reborn", "revive", spell);  
  
    //Returns the amount of MagicCards in the deck  
    d.getMagicCardCount(); //1  
  
    //Returns the amount of MonsterCards in the deck  
    d.getMonsterCardCount(); //13  
  
    Duelist player1("Seto Kaiba");  
  
    ///Changes the 2-nd monster card in the Monster deck to "Dark Magician"  
    MonsterCard DarkMagician("Dark Magician", 2500, 2100);  
    player1.changMonsterInDeck(2, DarkMagician);  
  
    Duelist player2("Yugi Muto");  
  
    //Changes the 2-nd magic card in the Magic deck to "Reflection"  
    MagicCard Reflection(Reflection, "Reflects the last played spell", trap);  
    Player2.changeMagicCardInDeck(2, Reflection );  
  
}
```

---





# ООП – Практикум: Домашна работа №1

## Пояснение:

- Опитайте се да напишете максимално ефективен код, както по отношение на брой редове, така и по отношение на време за изпълнение. Помислете къде може да се намали броят на повторенията на циклите или да се намали броят на променливите, които използвате за решаване на задачата. Реализирайте задачите спазвайки добрите ООП практики, за които се говори по време на лекции и упражнения.
- Файловете с решенията може да съдържат само стандартните символи с кодове от 0-127 (не се разрешава използване на кирилица, например в стринговете или коментарите!).
- Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.
- **Не е разрешено** да ползвате класове от библиотеката STL като `std::string`, `std::vector`, `std::stack` и др
- Първото нещо във всеки от файловете, които предавате, трябва да бъде коментарен блок, който носи информация за съдържанието на файла. Този коментар трябва да изглежда точно така, както е показано по-долу, като в него попълните информация за Вас. За улеснение, просто копирайте дадения по-долу блок и попълнете в него необходимите данни, вместо текста, маркиран с ъглови скоби. Обърнете внимание, че на първия ред след наклонената черта има две звезди и че във файловете не може да се съдържат символи на кирилица.

---

```
/**
 * Solution to homework assignment 1
 * Object Oriented Programming Course
 * Faculty of Mathematics and Informatics of Sofia University
 * Summer semester 2020/2021
 *
 * @author <вашето име>
 * @idnumber <вашият факултетен номер>
 * @task <номер на задача>
 * @compiler <използван компилатор - GCC или VC>
 */
```

---





# ООП – Практикум: Домашна работа №1

Например един попълнен блок за студент с име Иван Иванов, Ф.н:12345, който предава задача 2, компилирана с GCC, трябва да изглежда така:

---

```
/**  
 * Solution to homework assignment 1  
 * Object Oriented Programming Course  
 * Faculty of Mathematics and Informatics of Sofia University  
 * Summer semester 2020/2021  
 *  
 * @author Ivan Ivanov  
 * @idnumber 12345  
 * @task 2  
 * @compiler GCC  
 */
```

---

## Изисквания за предаване:

- Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно и при установено плагиатство ще бъдат анулирани.
- 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 12345);

Пример за .zip архив за домашно: 1\_SI\_1\_1\_12345.zip

- Архивът да съдържа само изходен код (.cpp и .h/.hpp 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;
- Качване на архива на посоченото място в Moodle;

Софийски университет "Св. Климент Охридски"

Факултет по математика и информатика

