

## Курсова работа - изисквания

Курсовата работа трябва да се реализира в три етапа, като за всеки етап има краен срок за предаване в мидъл. След края на третия етап ще бъде назначена дата за защита на проекта, в която студентът трябва да демонстрира функционалността на програмата, както и да отговаря на въпроси относно нейната реализация. В процеса на разработка всеки от вече предадените етапи подлежи на промяна (в случай, че това е необходимо).

**Задание – срок на предаване: 27.10.2023г.** Заданието трябва да описва функционалните изисквания, на които ще отговаря проектът, като те трябва да са съобразени с изискванията по етапи. За оригинални задания (различни от стандартното задание и неповтарящи се) се дават допълнителни 5 точки.  
**Оригиначните задания задължително трябва да включват UML Class диаграма с йерархия от класове, която ще бъде използвана за реализация на проекта.**

### Изисквания относно реализацията:

Проектът трябва да демонстрира следните технологии (разпределени по етапи), като за всяка от тях студентът трябва да изтъкне необходимост от използване по време на защитата:

### Етап 1 – срок на предаване: 10.11.2023г.

1. **Наследяване** - нужно е да се реализира йерархия от класове (задължително);
2. **Капсулация** (задължително);
3. **Полиморфизъм** (задължително);
4. **Виртуални методи** (задължително);
5. **Употреба на различни модификатори за достъп** (минимум public и private) (задължително);
6. **Свойства** (задължително);
7. **Делегати и събития** (1т).

## **Етап 2 – срок на предаване: 24.11.2023г.**

- 1. Потребителски интерфейс за работа с програмата, различен от конзолен** (три до осем различни форми или команди). Проектът трябва да демонстрира прехвърляне на данни между отделните форми (до 8т);
- 2. Употреба на специализирана програмна функционалност, взаимодействаща с изградената в етап 1 йерархия** (пр: Реализиране на графични обекти чрез System.Drawing, изчислителна обработка на информация чрез Math.NET, обработка на XML схеми чрез System.Xml и т.н.). Специализираната функционалност може да бъде предложена от студента в зависимост от неговите интереси и предпочитания. (до 6 т.)
- 3. Реализиране на йерархия от класове за операциите от потребителския интерфейс с възможност за добавяне на нови операции без промяна по съществуващия код** (до 6 т.)
- 4. Реализиране на история на операциите с възможност за отмяна или повторно изпълнение** (напр. команди Undo и Redo; възможност за дефиниране на "макроси" - последователност от действия, дефинирани от потребителя, които се изпълняват "с един бутон"; функционалност replay - запис и повторение на действията на потребителя) (до 6 т.)

## **Етап 3 – срок на предаване: 15.12.2023г.**

- 1. Съхранение на информация във файлове, посредством сериализация** (до 4т);
- 2. Използване на LINQ** (минимум 4 вида операции) (до 8т);
- 3. Отделяне на преизползваемата логика на проекта в библиотека и премахване на външните зависимости на библиотеката** (до 4т).

**При неспазване на срока за даден етап се отнемат 5 точки.**

Оценяване:

- от 7 до 10 точки - Среден 3 \*
- от 11 до 25 точки - Добър 4
- от 25 до 35 точки - Мн. добър 5
- над 35 точки - Отличен 6

\*Функционалността от етап 1, отбелязана като задължителна, трябва да бъде налична, за да бъдат разглеждани следващите етапи.

-----

-----

**Стандартно задание:** „Приложение за работа с графични фигури“.

Приложението да предоставя графичен интерфейс за работа (Windows Forms), като на основната форма (сцената) потребителят трябва да може да създава множество фигури (поне три различни вида). Приложението да дава възможност за взаимодействие с фигурите (избиране, преместване, изтриване и т.н.). Да има възможност за изчисление на лице на фигурите в зависимост от техните размери. Да има възможност за редактиране на всички параметри на фигурите (напр. да се променя радиуса на окръжност, да се променя позицията на връх на отсечка, да може да се променя цвета на фигурите).

#### **Етап 1:**

1. **Наследяване** – йерархия от класове за фигурите;
2. **Капсулация** – всяка фигура притежава координати, размери, метод за изчисление на лице, метод за изрисуване и т.н.;
3. **Полиморфизъм** – фигурите се съхраняват в една колекция в класа на сцената и операциите върху тях се извършват еднотипно;
4. **Виртуални методи** – за всяка фигура може да се изчисли лице, да бъде изрисувана върху сцената, да бъде преместена и т.н.
5. **Употреба на различни модификатори за достъп** – за координатите, размерите на фигурите, методите на фигурите и т.н..
6. **Свойства** – координатите, размерите на фигурите и т.н.
7. **Делегати и събития** – класове за отделните команди като преместване, редакция и т.н. могат да предоставят събития, които да се активират при OnClick и т.н.

#### **Етап 2:**

1. **Интерфейс за работа с програмата** – Windows Forms интерфейс с процорец за сцена (където се визуализират фигурите), команди за редакция на фигура (промяна на размери, избор на цвят за запълване, цвят за ограждане и т.н.), диалогови прозоречи за съхранение и зареждане на данните в/от файл;
2. **Употреба на специализирана програмна функционалност**  
**взаимодействаща с изградената в етап едно йерархия** – Използване на функционалността на System.Drawing за визуализация на фигурите. Всяка фигура може да се създава, изтрива, избира с мишката и премества (чрез влачене с мишката).

3. **Реализиране на йерархия от класове за операциите от потребителския интерфейс с възможност за добавяне на нови операции без промяна по съществуващия код** - интерфейс за команда, наследници за създаване, местене и други действия с фигурите.

4. **Реализиране на история на операциите с възможност за отмяна и/или повторно изпълнение** - Undo/Redo чрез възможност командите да се връщат назад и списъци с изпълнени и върнати назад команди

### **Етап 3:**

1. **Съхранение на информация във файлове, посредством сериализация** – сцената и създадените фигури могат да се запишат във файл, откъдето в последствие да бъдат заредени;
2. **Използване на LINQ** – при работа с колекцията с фигури на сцената;
3. **Реализация на библиотека** – йерархията за фигури да се реализира в библиотека, като фигурите са независими от Windows Forms класовете и няма зависимост към System.Drawing.