

# Vásárhelyi Máté – MTGI7S

## Programozás alapjai 2, Nagy Házi Feladat

### Alakzatok

#### 1.) Specifikáció

A programom egy absztrakt alakzat osztályt valósít meg. Ennek segítségével hozok létre négyzet, kör, illetve szabályos háromszög alakzatokat. Az alakzatokat középpontjukkal, illetve egy csúcsukkal lehet egyértelműen definiálni. (Kör esetén a körvonal egyik pontja). Ezen kívül a kör középponttal és sugárral, a háromszög és a négyszög pedig középponttal és oldalhosszal is létrehozható. Az alakzatok tartalmaznak metódust mellyel eldönthető hogy tartalmazza-e őket egy tetszőleges kör, illetve tartalmaznak metódust amivel eldönthető, hogy egy pont az alakzat területén belül található e. A program fájlból, illetve standard bemenetről is képes alakzatokat beolvasni, tárolni. Képes a tárolt alakzatokat szűrni az alapján, hogy tartalmazza-e őket egy kör, illetve hogy egy pontot tartalmaznak e. A program képes egy adott szűrés eredményeit eltárolni, illetve fájlba vagy standard kimenetre kiírni.

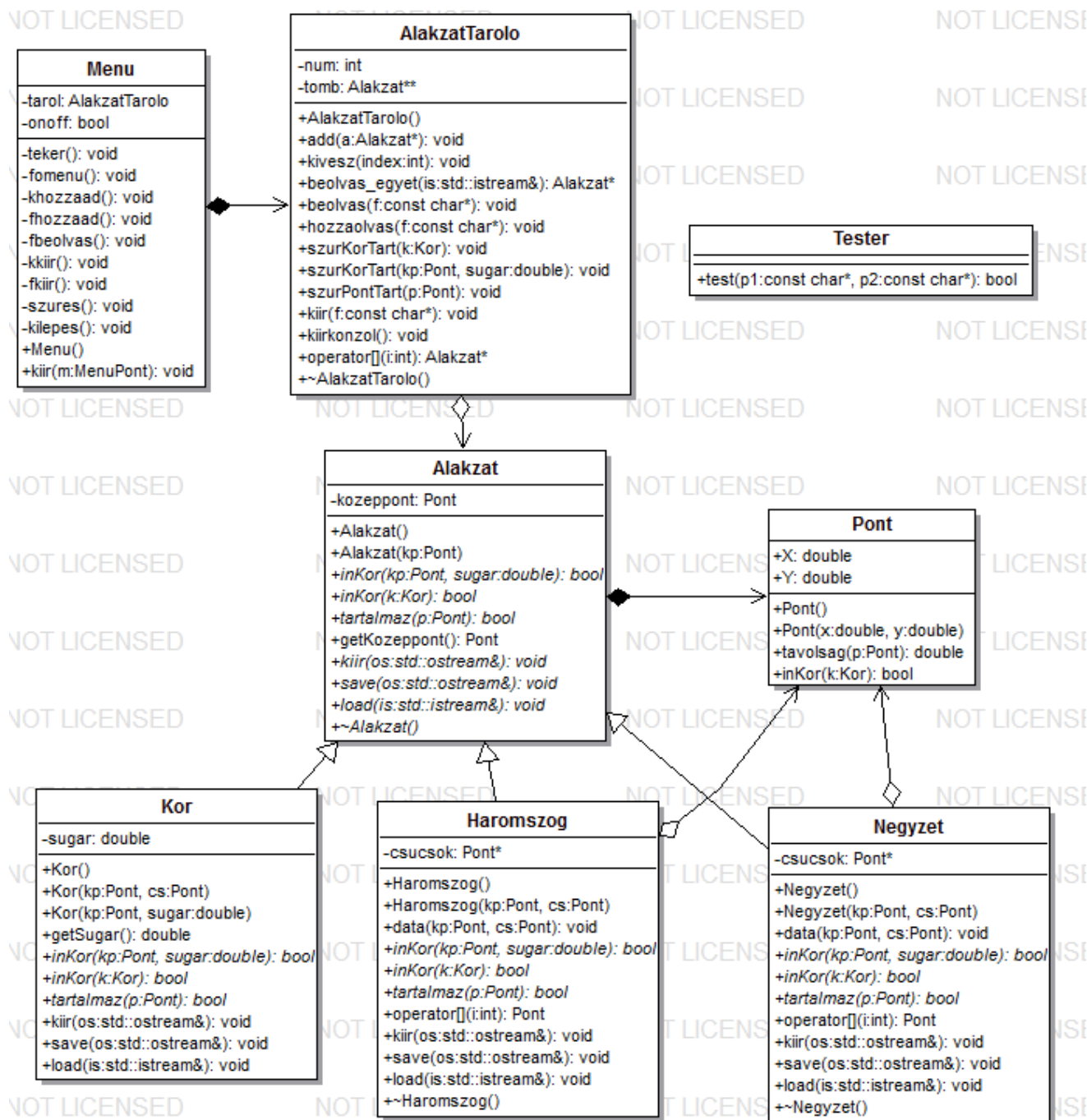
#### 2.) Megoldási vázlat

Először kitaláltam, hogy milyen függvények/változók kellenek minden alakzat osztályba, majd ennek alapján megterveztem az Alakzat absztrakt őosztályt. Ezután megterveztem egyesével a leszármaztatott osztályokat. Létrehoztam a Pont osztályt a csúcsok és távolságok könnyebben kezelhetősége érdekében. Ezután terveztem meg a dinamikus tárolót amiben az alakzatokat tudom tárolni, illetve kezelni. Végül megírtam a menüt, hogy felhasználóként lehessen kezelni a programot, majd egy gépi tesztelő osztályt.

#### 3.) Tesztelési dokumentáció

A memóriaszivárgást a már korábban is használt memtrace modul segítségével teszteltem. A program helyes működésének tesztelésére írtam egy tesztelő osztályt ami figyel, hogy adott bemenet és elvárt kimenet esetén az utóbbi egyezik-e a program által generált kimenettel. Ezután írtam egy bemenetet amely végighívja a programomban a függvények jelentős részét, és ezt futtattam le, amire a teszter sikeres tesztelést jelzett. Ezenkívül manuálisan is leteszteltem külön-külön a függvények működését.

## 4.) Osztálydiagram



## 5.) Osztályok és tagfüggvények rövid leírása (Doxygen)

### Alakzat Class Reference

```
#include <alakzat.h>
```

## Public Member Functions

- **Alakzat** (**Pont** kp)  
*Konstruktor megadott középponttal.*
- virtual bool **inKor** (**Pont** kp, double sugar)=0  
*Tisztán virtuális függvény. Megmondja, hogy az alakzat benne van-e egy körben.*
- virtual bool **inKor** (**Kor** k)=0  
*Tisztán virtuális függvény. Megmondja, hogy az alakzat benne van-e egy körben.*
- virtual bool **tartalmaz** (**Pont** p)=0  
*Tisztán virtuális függvény. Megmondja, hogy az alakzat tartalmaz-e egy pontot.*
- **Pont** **getKozepPont** () const  
*Középpont getter függvénye.*
- virtual void **kiir** (std::ostream &os) const =0  
*Tisztán virtuális függvény. Kiírja az alakzatot a kapott stream-re.*
- virtual void **save** (std::ostream &os) const =0  
*Tisztán virtuális függvény. Lementi az alakzatot a kapott stream-re.*
- virtual void **load** (std::istream &is)=0  
*Tisztán virtuális függvény. Betölti az alakzatot a kapott stream-ről.*
- virtual ~**Alakzat** ()  
*Virtuális destruktork.*

## Protected Attributes

- **Pont** **kozeppont**

## Detailed Description

Absztrakt alakzat ösosztály, ebből örökölnek a különböző alakzatok.

## Member Data Documentation

### **Pont** **Alakzat::kozeppont** [protected]

Az alakzat középpontja.

## AlakzatTarolo Class Reference

```
#include <tarolo.h>
```

## Public Member Functions

- void **add** (**Alakzat** \*a)  
*Alakzat hozzáadása.*
- void **kivesz** (int index)  
*Alakzat eltávolítása.*
- **Alakzat** \* **beolvas\_egyet** (std::istream &is)  
*Egy alakzat beolvasása az adott stream-ről.*
- void **beolvas** (const char \*f)  
*Az összes alakzat beolvasása fájlból.*

- void **hozzaolvas** (const char \*f)  
*Az összes alakzat eddigiekhez való hozzáadása az adott stream-ről.*
- void **szurKorTart** (**Kor** k)  
*A tárolt alakzatok szűrése az alapján, hogy benne vannak-e egy körben.*
- void **szurKorTart** (**Pont** kp, double sugar)  
*A tárolt alakzatok szűrése az alapján, hogy benne vannak-e egy körben.*
- void **szurPontTart** (**Pont** p)  
*A tárolt alakzatok szűrése az alapján, hogy tartalmaznak-e egy pontot.*
- void **kiir** (const char \*f)  
*A tárolt alakzatok kiírása fájlba.*
- void **kiirkonzol** ()  
*A tárolt alakzatok kiírása konzolra.*
- **Alakzat \* operator[]** (int i)  
*Indexelő operátor, visszaadja az adott alakzatot.*
- **~AlakzatTarolo** ()  
*Destruktor.*

## Detailed Description

Alakzattároló osztály. Dinamikusan tárol és kezel alakzatokat.

## Haromszog Class Reference

```
#include <haromszog.h>
```

### Public Member Functions

- **Haromszog** ()  
*Alapértelmezett konstruktor.*
- **Haromszog** (**Pont** kp, **Pont** cs)  
*Konstruktor középponttal és egy csúccsal.*
- void **data** (**Pont** kp, **Pont** cs)  
*Háromszög középpontjából és egy csúcsából a többi csúcs kiszámítása.*
- virtual bool **inKor** (**Pont** kp, double sugar)  
*Megmondja, hogy a háromszög benne van-e egy körben.*
- virtual bool **inKor** (**Kor** k)  
*Megmondja, hogy a háromszög benne van-e egy körben.*
- virtual bool **tartalmaz** (**Pont** p)  
*Megmondja, hogy a háromszög tartalmaz-e egy pontot.*
- **Pont operator[]** (int i) const  
*Indexelő operátor visszaadja a háromszög adott csúcsát.*
- void **kiir** (std::ostream &os) const  
*Kiírja a háromszöget a kapott stream-re.*
- void **save** (std::ostream &os) const  
*Lementi a háromszöget a kapott stream-re.*
- void **load** (std::istream &is)  
*Betölti a háromszöget a kapott stream-ről.*

- **~Haromszog ()**  
*Destruktor.*
- 

## Detailed Description

Háromszög osztály, Alakzattól származtatott.

## Kor Class Reference

```
#include <kor.h>
```

### Public Member Functions

- **Kor (Pont kp, Pont cs)**  
*Alapértelmezett konstruktor.*
  - **Kor (Pont kp, double sugar)**  
*Konstruktor középponttal és sugárral.*
  - **double getSugar ()**  
*A sugár getter függvénye.*
  - **virtual bool inKor (Pont kp, double sugar)**  
*Megmondja, hogy a kör benne van-e egy körben.*
  - **virtual bool inKor (Kor k)**  
*Megmondja, hogy a kör benne van-e egy körben.*
  - **virtual bool tartalmaz (Pont p)**  
*Megmondja, hogy a kör tartalmaz-e egy pontot.*
  - **void kiir (std::ostream &os) const**  
*Kiírja a kört a kapott stream-re.*
  - **void save (std::ostream &os) const**  
*Lementi a kört a kapott stream-re.*
  - **void load (std::istream &is)**  
*Betölti a kört a kapott stream-ről.*
- 

## Detailed Description

Kör osztály, Alakzattól származtatott.

## Menu Class Reference

```
#include <menu.h>
```

### Public Member Functions

- **Menu ()**  
*Konstruktor.*
- **void kiir (MenuPont m)**  
*Kiír egy adott menüpontot.*

---

## Detailed Description

Menü osztály.

## Negyzet Class Reference

```
#include <negyzet.h>
```

### Public Member Functions

- **Negyzet ()**  
*Alapértelmezett konstruktor.*
- **Negyzet (Pont kp, Pont cs)**  
*Konstruktor középponttal és egy csúccsal.*
- **void data (Pont kp, Pont cs)**  
*Negyzet középpontjából és egy csúcsából a többi csúcs kiszámítása.*
- **virtual bool inKor (Pont kp, double sugar)**  
*Megmondja, hogy a négyzet benne van-e egy körben.*
- **virtual bool inKor (Kor k)**  
*Megmondja, hogy a négyzet benne van-e egy körben.*
- **virtual bool tartalmaz (Pont p)**  
*Megmondja, hogy a négyzet tartalmaz-e egy pontot.*
- **Pont operator[] (int i) const**  
*Indexelő operátor visszaadja a négyzet adott csúcsát.*
- **void kiir (std::ostream &os) const**  
*Kiírja a négyzetet a kapott stream-re.*
- **void save (std::ostream &os) const**  
*Lementi a négyzetet a kapott stream-re.*
- **void load (std::istream &is)**  
*Betölti a négyzetet a kapott stream-ről.*
- **~Negyzet ()**  
*Destruktor.*

---

## Detailed Description

Négyzet osztály, Alakzathól származtatott.

## Pont Class Reference

```
#include <pont.h>
```

### Public Member Functions

- **Pont ()**  
*Alapértelmezett konstruktor.*

- **Pont** (double x, double y)  
*Konstruktor koordinátákkal.*
- double **tavolsag** (**Pont** p)  
*Megadja egy másik ponttól való távolságát a pontnak.*
- bool **inKor** (**Kor** k)  
*Megmondja hogy a pont benne van-e egy körben.*

### Public Attributes

- double **X**
- double **Y**

## Tester Class Reference

```
#include <tester.h>
```

### Public Member Functions

- bool **test** (const char \*p1, const char \*p2)  
*Két fájl tartalmának egyezését vizsgáló függvény.*

---

### Detailed Description

Tesztter osztály. Gépi tesztelést hajt végre.