

Paraméterek beolvasása

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2021. szeptember 15.

Emlékeztető: az első program

Másodfokú egyenlet megoldóképlete

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double a, b, c;
6     double d, r1, r2;
7     a = 1.0;
8     b = 3.0;
9     c = 2.0;
10    d = b * b - 4 * a * c;
11    d = sqrt(d);
12    r1 = (-b + d) / (2 * a);
13    r2 = (-b - d) / (2 * a);
14    printf("r1 = %f, r2 = %f\n", r1, r2);
15    return 0;
16 }
```

Az előző program egy kicsit tömörebben

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double a = 1.0;
6     double b = 3.0;
7     double c = 2.0;
8     double d = sqrt(b * b - 4 * a * c);
9     double r1 = (-b + d) / (2 * a);
10    double r2 = (-b - d) / (2 * a);
11    printf("r1 = %f, r2 = %f\n", r1, r2);
12    return 0;
13 }
```

- a változók deklarálása összevonható az értékadással

Az előző program egy kicsit tömörebben

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double a = 1.0;
6     double b = 3.0;
7     double c = 2.0;
8     double d = sqrt(b * b - 4 * a * c);
9     double r1 = (-b + d) / (2 * a);
10    double r2 = (-b - d) / (2 * a);
11    printf("r1 = %f, r2 = %f\n", r1, r2);
12    return 0;
13 }
```

- a változók deklarálása összevonható az értékadással
- a függvények kifejezéseket is kaphatnak paraméterként
 - ilyenkor a kifejezés előbb kiértékelődik
 - a függvény a kapott eredményt kapja meg paraméterként

- 1 Az egyenlet együtthatói nem paraméterek, hanem konstansok
 - ez úgy mondjuk, hogy az értékek “hard code”-olva vannak
 - az ilyet mindig kerülni kell, kivéve ha valóban konstansokról van szó

- 1 Az egyenlet együtthatói nem paraméterek, hanem konstansok
 - ez úgy mondjuk, hogy az értékek “hard code”-olva vannak
 - az ilyet mindig kerülni kell, kivéve ha valóban konstansokról van szó
- 2 Hogyan lehetne az együtthatókat paraméterként beadni?
 - parancssori argumentumként
 - billentyűzetről beolvasva
 - fájlból beolvasva

A parancssori argumentumokat a konzolablakban szeretnénk megadni futtatáskor, pl.:

```
1 $ ./roots 1 3 2
```

A parancssori argumentumokat a konzolablakban szeretnénk megadni futtatáskor, pl.:

```
1 $ ./roots 1 3 2
```

Ezeket az argumentumokat a `main` függvény paramétereiként kapjuk meg

- vigyázat: minden paraméter szöveggént van kezelve
- át kell alakítani a szöveget számmá
- egyelőre itt a kész megoldás: `atof` függvény

Parancssori paraméterek beolvasása

A parancssori argumentumokat a konzolablakban szeretnénk megadni futtatáskor, pl.:

```
1 $ ./roots 1 3 2
```

Ezeket az argumentumokat a `main` függvény paramétereiként kapjuk meg

- vigyázat: minden paraméter szöveggént van kezelve
- át kell alakítani a szöveget számmá
- egyelőre itt a kész megoldás: `atof` függvény

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <math.h>
4
5 int main(int argc, char* argv[]) {
6     double a = atof(argv[1]);
7     double b = atof(argv[2]);
8     double c = atof(argv[3]);
9     double d = sqrt(b * b - 4 * a * c);
10    double r1 = (-b + d) / (2 * a);
11    double r2 = (-b - d) / (2 * a);
12    printf("r1 = %f, r2 = %f\n", r1, r2);
13    return 0;
14 }
```

- Az `argc` paraméter a parancssori paraméterek számát tartalmazza
 - az első (0. indexű) paraméter mindig a futó program neve
- Az `argv` paraméterben szöveggént kapjuk meg a paramétereket