Másoló-mozgató szemantika

1. **Másoló szemantika**:

Ha van az osztályban dinamikus adattag, akkor alapértelmezésben tiltjuk , míg privátban deklaráljuk a másoló konstruktort és másoló értékadást.Különben alkalmazzuk a “Rule of three” szabályt.

A Rule of three szabály kimondja, hogy ha dinamikus tagot aggregáló osztályban kell implementálni:

* Destruktort,
* Másoló konstruktort,
* Vagy másoló értékadást

Akkor nagy valószínűséggel mindhármat implementálni kell.

Példa Rule of three-re:

Int(const Int&);

Int(Int&&); //Rule of 5

Int& operator=(const Int&);

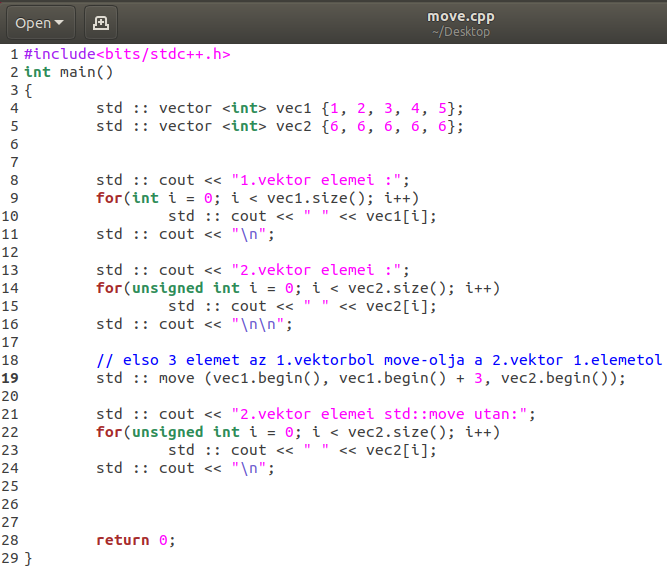
Int& operator=(Int&&);

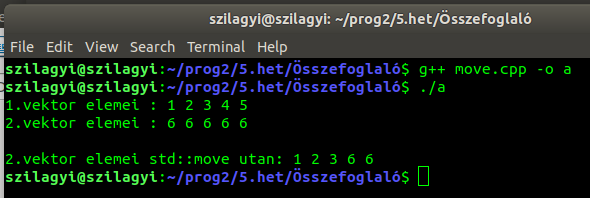
~Int();

2.**Mozgató szemantika:**

C++11-be bekerűlt a mozgató konstruktor, amely alkamazása drasztikusan hatékonyabbá tette a kódokat. A mozgató konstruktor legfontosabb tulajdonsága, hogy a használatával elkerülhető a memória újraelosztása és ugyanannyi memóriát használ mint amennyit az eredeti objektum használt, aminek ideiglenes funkciója van, mivel miután átadta az értéket az új objektumnak törlésre kerül. Ahhoz, hogy alkalmazni tudjuk a mozgató szemantikát, először meg kell oldanunk azt a problémát, hogy az eredeti objektumot hibamentesen töröljük.

Példa:





**Copy ctor** vs **Move ctor**

|  |  |
| --- | --- |
| Másoló konstruktor | Mozgató konstruktor |
| Az argumentum referenciaként a bal oldali értéket kapja. | Az argumentum referenciaként a jobb oldali értéket kapja. |
| Új objektumot hoz létre a kapott objektumból az által, hogy lemásolja az összes értékét egy új memória címre. | Új objektumot hoz létre, de ugyanannyi memóriát használ mint az átadott objektum. |
| Nem hatékony, mivel sok memória elosztást használ működése során az új objektumok létrehozására.A mozgató konstruktor használata sokkal hatékonyabb. | Memóriahasználatának nagy részét a kapott objektum teszi ki, ezért hatékonyabb, mint a másoló konstruktor. |
| Nem változtat semmit a kapott objektumon, ezért a kapott objektum használható másoló operátor után is. | Mivel a mozgató konstruktor a kapott objektum memória blokkjait használja, ezért a kapott objektum nem használható mozgató operátor után. |