

BB

Programozás alapjai II. 3. ellenőrző dolgozat. 2012.04.05. Kurz/Terem: G3/	20 perc
Név:	Neptun:
Összpont:	

1. feladat

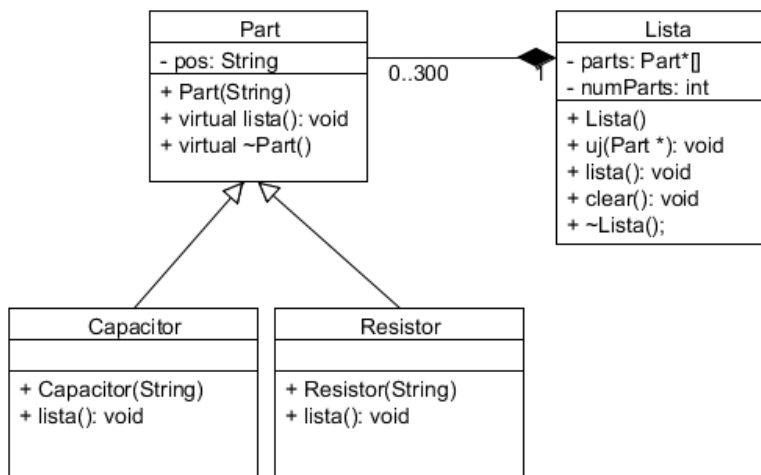
3 pont

Elektronikus áramkörök tervezését támogató rendszerben az **alkatrészlistát** úgy szeretnénk elkészíteni, hogy minden alkatrész mellé kirajzoljuk az áramkör jelét is. Minden alkatrésznek van egy szöveges pozíciójele is (pl: R12). Tudjuk, hogy az alkatrészek száma soha nem haladja meg a 300-at. Először csak ellenállást (*Resistor*) és kondenzátort (*Capacitor*) akarunk használni, de később újabb alkatrészekkel akarjuk bővíteni a rendszert. Az alkatrészek (*Part*) pozíciójelét *String*-ként kell tárolni. A rendszerben megvalósítandó műveleteket egy kódrészlettel mutatjuk be:

```

Lista elemek; // Ez lesz az alkatrészlista
elemek.uj(new Resistor("R1")); // R1 jelű ellenállás hozzáadása a listához
elemek.uj(new Capacitor("C4")); // C4 jelű kondenzátor hozzáadása a listához
elemek.lista(); // alkatrészlista kiírása
elemek.clear(); // alkatrészlista törlése
    
```

Feltételezheti, hogy a *Lista* osztályból példányosított objektumot nem akarjuk paraméterként átadni és értékadás jobb, ill. bal oldalán sem szerepel. **Tervezz** meg és **rajzoljon** fel egy olyan osztályhierarchiát, ami alkalmas az alkatrészek tárolására és könnyen bővíthető. Az osztálydiagramban jelölje az adattagok és metódusok láthatóságát is! A *String* osztályt nem kell lerajzolni, arra típusként hivatkozzon!



2. feladat

3 pont

Deklarálja a *Lista*, *Part*, és *Resistor* osztályokat! **Csak** a *Lista* és a *Resistor* osztályok konstruktorát, valamint a *Lista* osztály *uj()*, és *lista()* metódusát **valósítsa** meg!

```

class Part {
    String pos;
public:
    Part(String);
    virtual void lista();
    virtual ~Part();
};

class Resistor : public Part
{
public:
    Resistor(String s)
        :Part(s) {}
    void lista();
};
    
```

```

class Lista {
    Part* parts[300];
    int numParts;
public:
    Lista() :numParts(0) {}
    void uj(Part* p) {
        parts[numParts++] = p;
    }
    void lista() {
        for (int i = 0; i < numParts; i++)
            parts[i]->lista();
    }
    void clear();
    ~Lista();
};
    
```