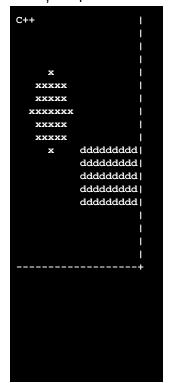
Problema 3:

Scrieti codul necesar pentru ca următoarea functie main()

```
int main() {
    //Canvas c; // ACEST COD NU VA COMPILA
    Canvas c(20, 20);
    c.AddPainter("cerc", new Circle());
    c.AddPainter("rect", new Rectangle());
    c(0, 0) = 'C'; c(1, 0) = '+'; c(2, 0) = '+';
    c.Paint("cerc", 5, 7, 3, 'x'); // cerc x=5,y=7, raza=3, fill='x'
    c.Paint("rect", 10, 10, 18, 14, 'd'); // dreptunghi (10,10) - (18,4) - fill='d'
    c.Paint("rect", 0, 19, 20, 19, '-');
    c.Paint("rect", 19, 0, 19, 19, '|');
    c(19, 19) = '+';
    for (int y = 0; y < c.GetHeight(); y++) {
        for (int x = 0; x < c.GetWidth(); x++)
            cout << c(x, y);
        cout << endl;</pre>
    return 0;
}
```

să afiseze pe ecran următoarele:



Observatii:

 Clasele Circle şi Rectangle sunt derivate dintr-o clasa abstractă comună IPainter, definită în felul următor:

Clasa Canvas conține un *map* cu cheie de tipul string și valoare de tipul pointer către o astfel de interfata.

- Initializarea unui canvas se face furnizand doi parametri: width si height. La inițializare se crează o matrice cu elemente de tipul char având acele dimensiuni. Operatorul apel funcție returnează o referință către un element din matrice și primește parametri o coordonata X și una Y.
- Asigurați-vă ca specificațiile sunt respectate (ex: codul care nu ar trebui sa compileze, NU compilează).

Barem:

- [5p] Fiecare clasa este scrisa folosind un fișier header și unul cpp. Codul din main este într-un fișier separat.
- [2p] Metoda AddPainter
- [2p] Metodele GetHeight si GetWidth
- [4p] Metoda Paint din clasa Canvas
- [6p] operatorul de apel de functie
- [6p] clasa Rectangle derivata din IPainter
- [10p] clasa Circle derivata din IPainter