# Задължително домашно #3

Да се напише програма за симулация на структурата на град.
Градът се представя като двумерна матрица от клетки. Всяка клетка може да бъде празна, улица или сграда. В града има различни видове сгради с различни функции.

1. Дефинирайте клас `City` за град, който има двумерна матрица с клетки.
    a. дефинирайте 1 публичен конструктор, който приема два целочислени аргумента за размерите на матрицата и я инициализира с празни клетки.
    b. дефинирайте метод `float getMaintenanceCost()`, който връща общата цена за поддръжка на всички клетки
    c. дефинирайте метод `int getRoadLength()`, който връща общия брой на всички клетки с улици
    d. дефинирайте метод `int getResidentCapacity()`, който връща общия брой жилищни места в клетките с домове
    e. дефинирайте метод `void build(unsigned int x, unsigned int y, Cell* cell)`, който поставя дадената клетка на подадените координати в матрицата. Преди поставяне методът трябва да провери да са спазени изискванията за тази клетка
    f. дефинирайте метод `float getRequiredPower()`, който връща нужната електроенергия за целия град
    g. дефинирайте метод `int getHappiness()`, който връща общото щастие за целия град
2. Дефинирайте базов клас `Cell` за клетка от града. В него поставете общите атрибути и методи на всички клетки. *Изберете механизъм, чрез който да можете да различавате различните разновидности на клетки **ИЛИ** това различаване да не бъде нужно.*
3. Дефинирайте клас `Road` за улица. Всяка улица:
    a. има дължина 1
    b. цената за поддръжка е 1
    c. използва 1 електроенергия
    d. няма изисквания за поставяне
    e. при смятане на щастие пътищата не се броят за сгради
4. Дефинирайте клас `Residential` за жилищна сграда. В тези сгради живеят жителите на града.
    a. дефинирайте метод `int getResidentCapacity()`, който връща броя жилищни места в тази сграда
    b. задължително трябва да се постави до улица
5. Дефинирайте следните типове жилищни сгради:
    a. `LightResidential`
        i. поддържа 5 жителя
        ii. цената за поддръжка е 2
        iii. използва 2 електроенергия
    b. `MediumResidential`

        i. поддържа 20 жителя
        ii. цената за поддръжка е 4
        iii. използва 4 електроенергия
    c. `DenseResidential`
        i. поддържа 50 жителя
        ii. цената за поддръжка е 6
        iii. използва 8 електроенергия
        iv. до нея освен улица трябва да има и сграда от тип Leisure
6. Дефинирайте клас `Leisure` за сграда за почивка. Тези сгради увеличават щастието на града.
    a. дефинирайте метод `int getHappiness()`, който връща щастието от тази сграда
7. Дефинирайте следните типове сгради за почивка:
    a. `Mall`
        i. цената за поддръжка е 8
        ii. използва 4 електроенергия
        iii. дава по 1 щастие за всяка жилищна сграда, до която се допира
    b. `Park`
        i. цената за поддръжка е 4
        ii. използва 2 електроенергия
        iii. дава щастие за всяка жилищна сграда, до която се допира, според нейния тип: 1 за `LightResidential`, 2 за `MediumResidential`, 3 за `DenseResidential`
8. Дефинирайте клас `Industrial` за производствена сграда. Тези сгради намаляват щастието на града.
    a. дефинирайте метод `int getHappiness()`, който връща щастието от тази сграда. Резултатът винаги ще бъде отрицателно число.
9. Дефинирайте следните типове производствени сгради:
    a. `LightIndustrial`
        i. цената за поддръжка е 4
        ii. използва 4 електроенергия
        iii. дава -1 щастие за всяка жилищна сграда, до която се допира
    b. `DenseIndustrial`
        i. цената за поддръжка е 6
        ii. използва 8 електроенергия
        iii. дава -2 щастие за всяка не-производствена сграда, до която се допира
10. Общи
    a. спазвайте правилна енкапсулация
    b. добавете обработка на грешки когато се добавя сграда на координати извън размера на матрицата
    c. демонстрирайте в main работата на класовете и методите
    d. където е нужно и не е уточнено изрично може да добавяте конструктори, деструктори, методи, атрибути, предефинирани оператори

e. всички класове трябва да бъдат дефинирани в **различни файлове**. Изключение се прави за вложени класове и класове за изключения, ползвани само от един клас. За компилация може да използвате make и мейкфайл
f. при проверяване за съседство се зачитат всички 8 околни клетки