Cache eviction policy

Time Duration: 1:45hrs

Design CacheEvictionPolicy with 2 strategy LRU(Least recently used), MRU(Most recent used)
Notes:

- Cache size = 5
- Methods exposed to user:
    - Insert(key, value) - consider key, value as integer
    - Get(key) returns value
    - StateOfCache() return [{key, value}]

Example: LRU
Input: A B C D E F B G

1. A
2. A B
3. A B C
4. A B C D
5. A B C D E
6. F B C D E
7. F B C D E
8. F B G D E

Example: MRU
Input: A B C D E F C G B

1. A
2. A B
3. A B C
4. A B C D
5. A B C D E
6. A B C D F
7. A B C D F
8. A B G D F
9. A B G D F

Requirements:

- It should be extendable to accommodate other eviction strategies easily.
- Configurable cache size.
- Can use anything to structure the code: Classes/Structs.

Few code design principles:

- Modularity of code.
- Naming conventions.
- SOLID principles.

Evaluation Criteria:

- No UI, No external database, No external library is required.
- Keep everything in memory. *Mention your assumptions of data*.
- Focus on code design and correctness.

Submit a zip file of your code.