

Inlämningsuppgift 2

Inledning

Denna uppgift belyser **Hashtabell (eng Hash table)**. Du kommer att implementera en Hashtabell vilken använder sig av linjär sondering (linear probing). Hashtabellen ska användas som en ordlista med engelska ord (se beskrivning längre fram i dokumentet).

På kursens hemsida finner du följande filer vilka du behöver för att implementera en lösning (se beskrivning längre fram):

- HashTable.h
- Hash.h
- Element.h
- engWords.txt

Det finns även filer som kan användas för testning:

- TestElement.h
- TestingHashTable.cpp

Krav på uppbyggnad

Gränssnittet för hashtabellen ges av klassmallen `HashTable` enligt:

```
template <typename HashElement>
class HashTable
{
private:
    // member variables for you to decide on
    // but it is required that you use an array (not a vector) for the table
    int hashTableSize;
    unsigned int myHash(const HashElement& elem) const
    {
        static Hash<HashElement> hashFunc;
        return hashFunc(elem) % hashTableSize; // hashFunc(elem) is a call of the
                                                // defined operator () for HashElement
    }
public:
    HashTable(int hashTableSize = 101);
    HashTable(const HashTable& aTable) = delete;
    virtual ~HashTable();
    HashTable& operator=(const HashTable& aTable) = delete;
    int contains(const HashElement& elem) const; // returns index or -1
    bool insert(const HashElement& elem);
    bool remove(const HashElement& elem);
    // below are functions that must be defined but is not needed for the word-list
    const HashElement& get(int index) const;
    void makeEmpty();
    double loadFactor() const;
    int getNrOfElements() const;
    int getHashTableSize() const;
};
```

Hashnyckeln ges av parentes-operatorn (operatörn ()) genom functions-objekt-mallen (function object template) Hash. Denna ska inte modifieras!

```
template <typename Key>
class Hash
{
public:
    unsigned int operator()(const Key & key) const;
};
```

De element som ska placeras i hashtabellen ges av strukturen för klassen Element inklusive specialiseringen av mallen Hash enligt nedan. (Observera att Element endast är ett tillfälligt namn, du byter detta till den klass vars objekt ska hashas i hashtabellen):

```
class Element
{
private:
    // member variables
public:
    // constructors, destructor ... member functions
    // definition of == operator and != operator
};

template<>
class Hash<Element>
{
public:
    unsigned int operator()(const Element& elem) const
    {
        // to be implemented
        // the integer returned is calculated from data in elem

        return 0; // replace 0 with the calculated value
    }
};
```

Instruktioner

Hash-element

Det ska vara möjligt att placera engelska ord i hashtabellen. De engelska orden definieras av klassen EngWord vilken du ska definiera utifrån stommen i Element.h. Objekt av klasstypen EngWord ska innehålla ett engelskt ord som en sträng.

Du bestämmer hur hashfunktionen beräknar hashvärdet för klasstypen EngWord. Detta definieras i operatorn(). Se specialiseringen av klassmallen Hash i samband med beskrivningen av stommen i Element.h.

Hashtabellen

Använd HashTable.h och deklarerar de variabler som krävs för att hantera kollisioner genom linjär sondering (linear probing).

Dessutom ska konstruktor och destruktor samt övriga medlemsfunktioner definieras.

Ordlista - ett program som använder Hashtabell

Implementera i en fil benämnd HashingEnglishWords.cpp följande:

- En funktion vilken läser de engelska orden från fil och skapar/fyller en array med objekt av typen EngWord för de ord som lästs. Filens namn är engWords.txt och sökvägen till filen ska vara C:/temp/.
- En funktion som placerar alla EngWord-objekt i en hashtabell. Funktionen ska dessutom presentera belastningsgrad (load factor) som en utskrift.
- En funktion i vilken användaren matar in en följd av engelska ord (som en mening) och får besked om vilka ord som inte ingick i ordlistan. De ord som inte ingick i ordlistan ska presenteras som en utskrift.
- En funktion i vilken användaren matar in engelska ord som läggs in i hashtabellen. Om ett ord som användaren vill lägga in redan finns i tabellen ska detta meddelas och ordet läggs inte in i tabellen.
- En funktion i vilken användaren matar in engelska ord som ska tas bort från hashtabellen. Om ett ord som användaren vill ta bort inte finns i hashtabellen meddelas detta.

I main-funktionen ska följande utföras:

- Orden läses från fil
- Orden placeras i hashtabellen
- En meny presenteras där användaren kan välja att
 - mata in en följd av ord för att få besked om vilka av orden i följden som inte finns i ordlistan (hashtabellen)
 - lägg in nya ord
 - ta bort ord
 - avsluta

Överst i filen skriver du en kommentar i vilken du motiverar storleken på hashtabellen, dvs antalet ingångar (entries) samt en motivering till definitionen av hashfunktionen för EngWord.

Övrigt som ska uppfyllas

Vid borttagning ska element som redan finns i tabellen inte flyttas.

Inlämning

Din inlämning består av följande filer

Hash.h

EngWord.h och EngWord.cpp om du inte definierar funktionerna inom klassdeklarationen

HashTable.h

HashingEnglishWords.cpp

samt

TestElement.h

TestingHashTable.cpp

Dina filer får inte innehålla kompileringsfel och filerna ska följa de instruktioner som beskrivits ovan. Vidare ska programmet vara fungerande i enlighet med beskrivningen av vad som ska utföras i main-funktionen.