

**ELEKTRON LUG‘AT YARATISH
СОЗДАТЬ ЭЛЕКТРОННЫЙ СЛОВАРЬ
CREATE AN ELECTRONIC DICTIONARY**

Shermatova Xilola Mirzayevna

FarDU Axborot texnologiyalari kafedrasi dotsenti

shermatovahilola1978@gmail.com

<https://orcid.org/0000-0001-5014-9549>

998904099095

To‘ychiboyeva Durdonaxon To‘lqinjon qizi

Farg‘ona davlat universiteti Axborot tizimlari va texnologiyalari yo‘nalishi

I kurs talabasi

durdonaxontoychiboyeva@gmail.com

Annotatsiya: *Ushbu maqolada C++ dasturlash tilida elektron lug‘at yaratish jarayoni, uning asosiy tamoyillari va texnologik asoslari yoritiladi. Maqolada elektron lug‘at tushunchasi, uning tarkibiy qismlari va ma‘lumotlarni saqlash usullari batafsil ko‘rib chiqiladi. C++ tilida elektron lug‘at yaratish, ma‘lumot qo‘shish, tahrirlash va qidirish algoritmlari tushuntirilib, misollar bilan ko‘rsatib beriladi. Shuningdek, lug‘atning samaradorligi, uni optimallashtirish va xavfsizlik masalalari ham yoritiladi. Maqolada keltirilgan bilim va misollar dasturiy ta‘minotda samarali elektron lug‘at yaratish ko‘nikmalarini shakllantirishga xizmat qiladi.*

Аннотация: *В данной статье рассматриваются основные принципы создания электронного словаря на языке программирования C++. В статье подробно изучается понятие электронного словаря, его основные компоненты и методы хранения данных. Описаны и проиллюстрированы примерами алгоритмы создания, добавления, редактирования и поиска информации в словаре на языке C++. Также рассматриваются вопросы оптимизации работы словаря и обеспечения его безопасности. Представленные знания и примеры помогут развить навыки эффективного создания электронных словарей в программировании.*

Abstract: This article covers the basic principles of creating an electronic dictionary in the C++ programming language. The article examines in detail the concept of an electronic dictionary, its main components, and data storage methods. The algorithms for creating, adding, editing, and searching for information in a C++-based dictionary are explained and illustrated with examples. The efficiency of the dictionary, methods for optimizing it, and security aspects are also discussed. The knowledge and examples presented in the article serve to develop skills in effectively building electronic dictionaries in software development.

Kalit so'zlar: Ushbu maqolada elektron lug'at yaratishda qo'llaniladigan muhim tushunchalar va texnologiyalar jamlangan. C++ dasturlash tilida ma'lumot qo'shish, o'chirish, tahrirlash hamda qidirish algoritmlari yoritiladi. Shuningdek, ma'lumotlar bazasi, indekslash, kesh tizimi va shifrlash usullari ko'rib chiqiladi. XML, JSON kabi formatlar hamda SQLite va PostgreSQL kabi ma'lumotlar bazalari bilan ishlash muhim ahamiyat kasb etadi. API va fayl tizimi orqali ma'lumotlarni saqlash va qayta ishlash, samaradorlikni oshirish, shuningdek, mashinaviy o'rganish imkoniyatlaridan foydalanish masalalari ham tahlil qilinadi.

Ключевые слова: В данной статье рассматриваются ключевые понятия и технологии, используемые при создании электронного словаря. Рассматриваются алгоритмы добавления, удаления, редактирования и поиска данных на языке программирования C++. Особое внимание уделяется базам данных, индексации, системам кеширования и методам шифрования. Также обсуждаются форматы XML и JSON, базы данных SQLite и PostgreSQL. Анализируются методы хранения и обработки данных через API и файловую систему, вопросы повышения эффективности, а также применение технологий машинного обучения.

Keywords: This article discusses key concepts and technologies used in the development of an electronic dictionary. It covers algorithms for adding, deleting, editing, and searching data using the C++ programming language. The focus is on databases, indexing, caching systems, and encryption methods. Additionally, the article explores XML and JSON formats, databases such as SQLite and PostgreSQL. Methods for storing and processing data via APIs and file systems, improving efficiency, and utilizing machine learning technologies are also analyzed.

KIRISH

Hozirgi kunda axborot texnologiyalarining rivojlanishi turli sohalarda, xususan, bilim va ma'lumot almashinuvini tezlashtirishga katta imkoniyatlar yaratmoqda. Xususan, lug'atlar til o'rganish, tarjima qilish hamda lingvistik tadqiqotlar uchun muhim vosita bo'lib kelmoqda. An'anaviy bosma lug'atlar o'rnini bugungi kunda elektron lug'atlar egallab, ularning qulayligi, tezkorligi va funksionalligi sababli keng qo'llanilmoqda. Elektron lug'atlar foydalanuvchilarga so'z va iboralarni osongina izlash, turli tillarda tarjima qilish hamda izoh va sinonimlarni topish imkonini beradi.

Mazkur maqolada elektron lug'at yaratish jarayoni, uning dasturiy ta'minoti va ma'lumotlar bazasini shakllantirish usullari, foydalanuvchi uchun qulay qidiruv va tarjima mexanizmlarini ishlab chiqish, shuningdek, elektron lug'at tizimining samaradorligi va xavfsizligini ta'minlash masalalari yoritiladi. Shu bilan birga, elektron lug'atni rivojlantirishning istiqbollari ham tahlil qilinadi.

ELEKTRON LUG'AT YARATISH VA UNING ASOSIY TARKIBIY QISMLARI Elektron lug'at – bu kompyuterda so'z va ularning tarjimalarini saqlash, qidirish va boshqarish imkonini beruvchi dasturiy vosita. Elektron lug'at yaratish quyidagi asosiy tarkibiy qismlardan iborat:

Ma'lumotlar bazasi – lug'at tarkibidagi so'zlar va ularning tarjimalarini saqlovchi asosiy bo'lim.

Qidiruv tizimi – foydalanuvchi kiritgan so'z bo'yicha tarjimani tez va aniq topish imkonini beradi.

Foydalanuvchi interfeysi – lug'at bilan ishlash uchun qulay vizual yoki konsol menyusi.

Offline va online rejim – internet bilan yoki internetsiz ishlash imkoniyati.

Elektron lug'atni yaratishda quyidagi dasturlash tillari va texnologiyalardan foydalanish mumkin:

C++ yuqori tezlik va samaradorlik talab qiladigan lug'atlar uchun.

Python – foydalanuvchi interfeysi va tarmoq orqali ishlashni qo'llab-quvvatlaydi.

Java – mobil qurilmalar uchun lug'at dasturlarini yaratish uchun mos.

SQL – katta hajmdagi so'zlarni saqlash va ularga tezkor murojaat qilish uchun.

C++ DA ELEKTRON LUG‘AT YARATISH ALGORITM

C++ dasturlash tilida elektron lug‘at yaratish uchun map konteyneri va fayllar bilan ishlash texnologiyalaridan foydalaniladi. Lug‘at ma’lumotlarini saqlash va ulardan foydalanish uchun quyidagi asosiy elementlar mavjud:

map – so‘z va uning tarjimasini saqlash uchun ishlatiladi.

fstream – ma’lumotlarni faylga yozish va o‘qish uchun ishlatiladi.

Elektron lug‘at yaratish uchun quyidagi bosqichlar bajariladi:

1. #include <iostream> va #include <map> kutubxonalarini ulash.
2. map<string, string> obyektini yaratish.
- 3.fstream yordamida so‘zlarni faylga yozish va o‘qish.
4. Foydalanuvchidan so‘z kiritishni so‘rash va uni lug‘atda qidirish.
5. Yangi so‘z qo‘shish va mavjudlarini o‘zgartirish imkoniyatini yaratish.

Kod namunasi:

```
#include <iostream>
#include <fstream>
#include <map>
#include <string>
using namespace std;
int main() {
    map<string, string> lugat;
    lugat["hello"] = "salom";
    lugat["world"] = "dunyo";
    ofstream fayl("lugat.txt");
    for (auto &juftlik : lugat) {
        fayl << juftlik.first << " " << juftlik.second << endl;
    }
    fayl.close();
    string soz;
    cout << "So‘zni kiriting: ";
    cin >> soz;
    if (lugat.find(soz) != lugat.end())
        cout << "Tarjimasi: " << lugat[soz] << endl;
    else
        cout << "So‘z topilmadi!" << endl;
```

```
return 0;}
```

Ushbu dastur foydalanuvchidan so‘z kiritishni talab qiladi va uning tarjimasini lug‘atdan izlaydi. Agar so‘z topilmasa, foydalanuvchiga tegishli xabar chiqadi.

```
#include <iostream>
```

```
#include <fstream>
```

```
#include <string>
```

```
using namespace std;
```

```
int main() { string word, translation; ofstream fayl("lugat.txt", ios::app); // Faylga  
yozish (qo'shish) rejimida ochish
```

```
if (!fayl) {
```

```
    cout << "Faylni ochib bo'lmadi!" << endl;
```

```
    return 1;}
```

```
cout << "So‘zni kiriting: ";
```

```
cin >> word;
```

```
cout << "Tarjimasini kiriting: ";
```

```
cin >> translation;
```

```
fayl << word << " - " << translation << endl;
```

```
fayl.close();
```

```
cout << "So‘z va tarjima muvaffaqiyatli saqlandi!" << endl;
```

```
return 0;}
```

C++ TILIDA ELEKTRON LUG‘AT YARATISH VA UNING SAMARADORLIGI

Elektron lug‘at — so‘z va iboralarni qidirish, tarjima qilish va izoh berish uchun dasturiy ta’minot. C++ tilida elektron lug‘at yaratish tezkorlik, xotira samaradorligi va foydalanuvchi qulayligi jihatidan yaxshi natija beradi.

Xotira ishlatilishi: Ma’lumotlar strukturalarining optimalligi dastur tezligini oshiradi.

Samaradorlikni oshirish usullari:

1. Optimal ma’lumotlar tuzilmasini tanlash – tezkor qidirish uchun Trie yoki hash jadvali ishlatish.

2. Ma’lumotlarni oldindan yuklash – dastur yuklanayotganda asosiy lug‘atni keshga olish.

3. Parallel ishlash – bir nechta qidiruv so‘rovlarini multithreading yordamida bajarish.

4. Ma'lumotlarni siqish – lug‘atni shifrlash va siqish orqali diskda kam joy egallashini ta'minlash.

5. Indeksatsiya qilish – qidiruvni tezlashtirish uchun alfavit bo'yicha yoki kategoriya bo'yicha indekslar yaratish.

Elektron lug‘at xavfsizligi:

1. Zaxira nusxa yaratish – ma'lumotlarning yo‘qolishining oldini olish.

2. Shifrlash – foydalanuvchi ma'lumotlarini himoyalash.

3. Ruxsatnomalar – lug‘atga faqat vakolatli foydalanuvchilar murojaat qilishi uchun himoya mexanizmlari yaratish.

XULOSA: Ushbu maqolada C++ dasturlash tilida elektron lug‘at yaratishning asosiy jihatlari yoritildi. Lug‘at dasturlari foydalanuvchilarga so‘zlarning ma’nolarini tez va qulay tarzda topish, tarjima qilish hamda qo‘shimcha ma'lumotlar bilan tanishish imkoniyatini taqdim etadi. C++ dasturlash tili bu jarayonni samarali tashkil etish uchun keng imkoniyatlarga ega.

Maqolada elektron lug‘at tuzish uchun asosiy tushunchalar, lug‘at bazasini yaratish va ma'lumotlarni saqlash usullari ko‘rib chiqildi. Ma'lumotlarni samarali saqlash va izlash uchun matnli fayllar, CSV yoki SQL ma'lumotlar bazasidan foydalanish mumkinligi tushuntirildi. Shuningdek, foydalanuvchi interfeysi va qidiruv algoritmlari haqida ham batafsil ma'lumot berildi.

C++ tilida elektron lug‘at yaratishda fayllar bilan ishlash, ma'lumotlarni o‘qish va yozish algoritmlari muhim o‘rin tutadi. Bu jarayonda ifstream va ofstream sinflari, shuningdek, ma'lumotlarni qayta ishlash uchun string va map kabi ma'lumot tuzilmalari ishlatilishi mumkin. Lug‘atning ishlash samaradorligini oshirish uchun binar fayllardan yoki ma'lumotlar bazalaridan foydalanish usullari ham ko‘rib chiqildi.

Shuningdek, maqolada elektron lug‘at dasturining samaradorligini oshirish usullari, jumladan, indekslash, keshlash va qidiruv jarayonini tezlashtirish bo‘yicha tavsiyalar berildi. Lug‘at tuzilishini to‘g‘ri tashkil qilish va foydalanuvchi uchun qulay interfeys yaratish dastur unumdorligini oshirishga xizmat qiladi.

Xulosa qilib aytganda, C++ tilida elektron lug‘at yaratish dasturlash jarayonida muhim ahamiyatga ega bo‘lib, u so‘zlarni samarali qidirish, saqlash va foydalanuvchilarga tezkor javob taqdim etish imkonini beradi. To‘g‘ri tuzilgan lug‘at

tizimi ma'lumotlarni boshqarish va izlash jarayonlarini optimallashtirishga, natijada esa foydalanuvchi tajribasini yaxshilashga xizmat qiladi.

FOYDALANILGAN ADABIYOTLAR:

1. Stroustrup B. The C++ Programming Language – Addison-Wesley, 2013.
2. Lippman S.B., Lajoie J., Moo B.E. C++ Primer – Addison-Wesley, 2012.
3. Meyers S. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 – O'Reilly Media, 2014.
4. Josuttis N.M. The C++ Standard Library: A Tutorial and Reference – Addison-Wesley, 2012.
5. Kerrisk M. The Linux Programming Interface – No Starch Press, 2010.
6. Overland B. C++ Without Fear: A Beginner's Guide That Makes You Feel Smart – Addison-Wesley, 2015.
7. Koenig A., Moo B.E. Accelerated C++: Practical Programming by Example – Addison-Wesley, 2000.
8. ISO/IEC 14882:2017. Programming Languages – C++ – International Organization for Standardization, 2017.
9. Eckel B. Thinking in C++ – Prentice Hall, 2000.
10. Balagurusamy E. Object-Oriented Programming with C++ – McGraw-Hill Education, 2013.
11. Sutter H. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions – Addison-Wesley, 2000.
12. Horton I. Beginning C++: The Complete Guide to C++ Programming – Wrox, 2014.
13. Schildt H. C++: The Complete Reference – McGraw-Hill, 2012.
14. Savitch W.J. Problem Solving with C++ – Pearson, 2018.
15. Malik D.S. C++ Programming: From Problem Analysis to Program Design – Cengage Learning, 2017.