

УДК 378.004.41+004.4

**Погромська Ганна Сергіївна**

доцент, кандидат педагогічних наук, доцент кафедри теорії й методики природничо-математичної освіти та інформаційних технологій,

Миколаївський обласний інститут післядипломної педагогічної освіти, Миколаїв, Україна

*hanna.pohromska@moippo.mk.ua*

ORCID: 0000-0002-6779-3995

**Махровська Наталя Анатоліївна**

кандидат фізико-математичних наук, доцент кафедри теорії й методики природничо-математичної освіти та інформаційних технологій,

Миколаївський обласний інститут післядипломної педагогічної освіти, Миколаїв, Україна

*natalya.makhrovska@moippo.mk.ua*

ORCID: 0000-0001-9603-6902

## **МЕТОДИКА ПОБУДОВИ ПРОГРАМНИХ КОМПЛЕКСІВ НА ПРИКЛАДІ ВІДКРИТОЇ КРОСПЛАТФОРМНОЇ ОНЛАЙН-СИСТЕМИ ДЛЯ ПЕРЕКЛАДУ ТЕКСТУ**

**Анотація.** Проаналізовано популярні онлайн-перекладачі Google Translate, Bing Translator, DeepL Translator і Apertium із метою порівняння їх інтерфейсів та функціональних можливостей. Досліджено переваги використання мови C++ та обґрунтовано вибір кросплатформного фреймворку Qt серед інших аналогів як такого, що задовольняє поставленим вимогам до розробки власного програмного продукту. Методологічну основу в аспекті розроблення програми складають методи об'єктно-орієнтованого програмування та використання інтерфейсу прикладного програмування. Представлено етапи розроблення та можливості онлайн-системи Crow Translate для перекладу тексту: інтерфейс прикладного програмування для перекладу, особливості реалізації програмної системи мовою C++ у фреймворку Qt, базові команди консольного інтерфейсу, функціональні можливості графічного інтерфейсу для озвучування та перекладу, варіанти вибору налаштувань, можливості локалізації та способи поширення пропонованої програми Crow Translate. У ході розроблення системи розділено реалізацію процесу перекладу та інтерфейсу програми на два окремих проєкта. Особливістю створеного проєкту є розробка окремої бібліотеки QOnlineTranslator для перекладу тексту, яка є вільним програмним забезпеченням та опублікована на сторінці GitHub під ліцензією GPL v3 для використання у будь-яких аналогічних проєктах. Описано деталі реалізації запропонованої бібліотеки та її основні функції. Висока якість перекладу у пропонованому продукті забезпечується завдяки використанню інтерфейсу Google. Передбачено можливість роботи в режимі словника під час уведення одного слова або фрагментів тексту. Для одного слова додатково відображаються різні варіанти перекладу та є підтримка відображення транслітерації. Під час розроблення програмного продукту враховано отримання перекладу у вікні, що спливає без додаткових переходів у браузер або інші додатки. У результаті отримано відкритий кросплатформний програмний продукт, який має зручний доступ до перекладу виділеного тексту в будь-якому додатку та потребує невеликого обсягу оперативної пам'яті.

**Ключові слова:** онлайн-перекладачі; розробка кросплатформної системи; C++; Qt; Google API; сирцевий код

**Вступ.** В умовах технічного прогресу людства щоденно збільшується обсяг світової комунікації. Сучасне міжкультурне спілкування є невід'ємною частиною як щоденної, так і професійної діяльності. Користувачам, які постійно працюють з іншомовними текстами,

необхідно застосовувати системи машинних перекладів, заснованих на різних технологіях доступу до словникових баз. Раніше це були програми, що склалися з програмного модуля і словників, які зазвичай доводилося завантажувати окремо. Проте більш зручними за умов наявності якісного Інтернету, є вебсервіси для перекладу, оскільки мають якісніший переклад та не потребують окремого завантаження словників. Загальний принцип роботи з такими системами полягає у використанні браузера для отримання перекладу, що не завжди є зручним. Швидка робота з текстовою інформацією та оперативний «just-in-time» переклад – потреба сучасного користувача. Тому розробка кросплатформної системи перекладу тексту на базі веб-сервісів машинного перекладу без необхідності додаткового переходу в інші системи є сучасною та актуальною.

**Аналіз літературних даних та постановка проблеми.** Проаналізовано наявні публікації щодо використання та розроблення онлайн-перекладачів. Роботи [1, 2] присвячені дослідженню онлайн-перекладачів Google Translate, PROMT та інших. Описано типові помилки машинного перекладу та визначено можливості і галузі застосування кожної з систем перекладу. У статтях [1, 2] зазначено, що машинні перекладачі доцільні для перекладу текстів із простими граматичними конструкціями, та зроблено висновок, що професійні перекладачі можуть використовувати ці системи лише для перекладу деяких типів тексту, щоб пришвидшити процес.

Стаття [3] висвітлює сучасні системи перекладу з наведенням їх основних характеристик, а також переваги та недоліки технологій машинного перекладу, які використовуються в цих системах.

Праці [4–6] присвячені огляду та аналізу найбільш продуктивних інтернет-ресурсів та додатків для вивчення не тільки основ англійської мови, але й лексики. Вивчено принципи роботи та склад новітніх електронних словників та отримано висновок про важливість використання сучасних технологій як в освітньому процесі, так і під час перекладу.

Статтю [7] присвячено визначенню переваг та недоліків деяких онлайн-перекладачів за критеріями, обраними авторами публікації.

У публікації [8] автори зазначають, що кожна система машинного перекладу має свої переваги. У своїй роботі дослідники об'єднали три системи: ієрархічну систему машинного перекладу, Bing Microsoft Translate і Google Translate, що дало змогу покращити якість перекладу. Автори оцінюють зазначену систему з погляду якості перекладу, але лише як користувачі.

Окреслені дослідження [1–8] описують здебільшого лінгвістичні основи машинного перекладу, інтерфейс наявних систем перекладу та особливості їх практичного застосування, але залишилися невирішеними питання, пов'язані з особливостями розроблення такого програмного забезпечення.

У роботі [9] описано функціонал мобільної навчальної системи та розглянуто наявні архітектури, зокрема MVC, та технології для розроблення таких систем. Автором обрано варіант продукту – вебдодаток для мобільних пристроїв, але таким чином, щоб користувач не звертався до браузера.

У дослідженні [10] представлена універсальна міжплатформна послуга обміну повідомленнями, в якій вирішена проблема кросплатформності служби передачі повідомлень. Авторами представлена підтримка передових технологій, включаючи IoT, та

застосовано інструменти RabbitMQ, Google-GSON та Quartz Scheduler для розроблення цієї системи. Вона може доставляти повідомлення незалежно від платформи обміну повідомленнями з доступними API, WebSockets з підтримкою HTML5 та додатків на основі TCP.

Автори роботи [11] пропонують критерії відбору технологій розроблення кросплатформних додатків. У статті [11] наведено детальний і ґрунтовний огляд інструментів розроблення кросплатформних додатків. Деякі з них дають всебічну оцінку, часто нехтуючи думкою програмістів про розробку міжплатформних додатків. Науковці Heitkötter H., Hanschke S., Majchrzak T. A. [11] зазначають, що більшість розробників програмного забезпечення зосереджуються на певних аспектах, таких, як анімація, продуктивність тощо. У дослідженні [11] наголошується на тому, що для створення додатків досі немає єдиних рекомендацій щодо використання веб-технологій, кросплатформного підходу або нативного програмного забезпечення (SDK). Ідеї роботи свідчать, що розроблення міжплатформних додатків зазнало значного прогресу, але кількість проблем постійно збільшується. Тому автори надають додаткову підтримку розробникам програмного забезпечення.

Зазначені роботи [9–11] розглядають підходи та можливості розроблення кросплатформного програмного забезпечення, надають рекомендації щодо вибору SDK, але не описують процес та деталі розробки додатків, зокрема онлайн-перекладачів.

Отже, продемонстровано, що внаслідок зростання популярності технологій розроблення кросплатформних систем із зазначеної теми підготовлено низку наукових праць [1–11]. Однак роботи [9–11] здебільшого не вирішують питання, пов'язані з розробкою програмного забезпечення, і обмежуються оглядом переважно однієї технології або описом критеріїв різних підходів до розроблення програмного продукту. Огляд періодичної літератури [1–11] дозволяє зробити висновок, що частина робіт [1–8] ґрунтується лише на способах застосування онлайн-перекладачів у професійній діяльності спеціалістів із погляду користувача. Проведений аналіз свідчить, що немає комплексних публікацій періодичних наукових видань, які присвячено способам і технологіям розроблення систем перекладу тексту. Причиною цього може бути той факт, що розробники програмного забезпечення, як правило, є не науковцями, а здебільшого – практикуючі програмісти.

Аналіз джерел [1–11] дозволяє стверджувати, що доцільним є проведення дослідження, що присвячене огляду інструментів для розроблення кросплатформних онлайн-додатків із акцентуванням уваги на описі деяких деталей реалізації онлайн-перекладача тексту в обраній системі. Під час опису процесу також отримуємо власний програмний продукт, який враховує окремі недоліки наявних нині аналогічних програмних систем перекладу текстів.

**Мета та задачі дослідження.** Метою роботи є вибір фреймворку для розроблення системи перекладу тексту та опис процесу реалізації такої програмної системи.

Для досягнення мети поставлено такі задачі:

1. Проаналізувати наявні онлайн-перекладачі за характеристиками платформ використання та функціональними можливостями.

2. Оглянути мови програмування та фреймворки щодо використання їх можливостей для зручної і швидкої розробки кросплатформного безкоштовного програмного

забезпечення.

3. Спроектувати систему, яка не залежить від використовуваної платформи і не потребує підключення додаткових інструментів для її роботи, та охарактеризувати ключові деталі реалізації онлайн-перекладача з використанням готових онлайн-баз граматичних конструкцій у розробленій бібліотеці.

4. Описати режими та особливості роботи розробленого програмного додатка.

**Методологічні основи та технологія створення кросплатформного програмного продукту для перекладу тексту.** Проаналізуємо сучасні програми-клієнти для популярних онлайн-сервісів перекладу тексту, які дозволяють отримати швидкий переклад потрібного фрагменту. Оглянемо можливості таких систем: Microsoft Translator (для сервісу Bing) [12], PROMT (для сервісу Translate.ru), TranslateClient від Олексія Ільїна [13], QTranslate від QuestSoft [14], Pragma від Trident Software [15] та Translate Shell від розробника з псевдонімом Siomort [16]. За результатами проведеного аналізу отримано зведену порівняльну таблицю (табл. 1).

Табл. 1  
Порівняння популярних програм-перекладачів

Назва	Операційна система	Відкритий сирцевий код	Безкоштовність	Інтерфейс взаємодії	Режим роботи
QTranslate	Windows	–	+	Графічний	on-line / off-line
PROMT	Windows	–	–	Графічний	on-line / off-line
Translate Shell	Linux, Windows	+	+	Консольний	on-line
Microsoft Translator	Windows	–	+	Графічний	on-line
TranslateClient	Windows	–	+/-	Графічний	on-line
Pragma	Windows	–	–	Графічний	off-line

Отже, на початок 2020 року для Linux не існує жодного сучасного перекладача з графічним інтерфейсом, а для Windows – жодного з відкритим сирцевим кодом. Крім того, деякі з них є комерційними проектами або мають обмежені функції у безкоштовній версії. Також окремі перекладачі підтримують переклад в off-line режимі, але для цього їх слід обов'язково встановити на комп'ютер.

Оглянемо сучасні мови програмування та фреймворки для створення програмних додатків та оберемо мову, яка максимально задовольняє вимоги до розробки власного програмного продукту. Було проаналізовано статистику найпопулярніших мов програмування для написання додатків від сайтів StrackOwerflow [17] та GitHub [18] (табл. 2).

Табл. 2

## Порівняння мов програмування

Мова програмування	Крос-платформність	Кросплатформні фреймворки для розробки графічного інтерфейсу	Тип мови
Java	Так	Swing, JavaFX	Інтерпретована
Python	Так	PyQt, Tkinter, WxPython, PyGtk	Інтерпретована
C#	Ні	–	Компільована
C++	Так	GTK, Qt, wxWidgets	Компільована
C	Так	-	Компільована
Ruby	Так	WxRuby, QtRuby, WxRuby, Shoes, TkRuby	Інтерпретована
Assembly	Ні	–	Компільована
Swift	Ні	–	Компільована
Objective-C	Ні	–	Компільована

Із табл. 2 видно, що такі мови як C#, Асемблер, Swift, Objective-C не відповідають вимогам розробки кросплатформних додатків, оскільки не надають відповідних засобів. Решта кандидатів – Ruby, Java і Python є інтерпретованими мовами, а C++ є компільованою мовою. Мовою програмування Java можна створювати міжплатформний програмний продукт без потреби додаткової зміни коду. Однак програми, реалізовані мовою Java, потребують більше ресурсів для своєї роботи, ніж інші. Програми, які скомпільовані в машинний код, працюють швидше та споживають менше ресурсів, ніж інтерпретовані. Тому для розробки перекладача, який задовольняє поставленим задачам, обрано мову C++.

Для швидкої розробки графічного інтерфейсу доцільно використовувати сучасний фреймворк Qt (кросплатформний фреймворк для розробки програмного забезпечення мовою програмування C++). Qt є основою багатьох програмних проєктів: QBittorrent, VLC, SMPlayer, Telegram Desktop, VirtualBox та ін., а також фундаментом популярного робочого середовища KDE, що входить до складу дистрибутивів Linux. Із усіх наведених у табл. 2 фреймворків найкраще задокументовано фреймворк Qt.

До переваг фреймворку Qt також можна віднести наявність унікальної системи сигналів і слотів. Вона повністю заміщує стару модель зворотного виклику та є гнучкою і повністю об'єктно-орієнтованою. Сигнали та слоти – це концепт програмування C++ із використанням Qt, який дозволяє поєднати непов'язані між собою об'єкти. Завдяки описаним перевагам для розроблення пропонованого програмного продукту був обраний фреймворк Qt.

Під час реалізації програми-перекладача було прийнято рішення розділити процес перекладу та інтерфейс програми на два окремих проєкти. Така модульність спрощує супровід коду і дозволяє використовувати механізм перекладу як окрему бібліотеку в інших проєктах.

Існує багато безкоштовних API для отримання перекладу, таких, як Google Translate,

Bing Translator, Yandex.Translate, DeepL Translator і Apertium. Зручною функцією програми є підтримка роботи відразу з декількома сервісами і можливість швидкого перемикання між ними, як це реалізовано в QTranslate [14].

Розроблена бібліотека з назвою QOnlineTranslator була опублікована на сторінці GitHub [19] під ліцензією GPL v3 [20]. Це дозволяє зручно відстежувати зміни, підключати бібліотеку як підмодуль у будь-який проєкт на GitHub та оперативно вносити зміни. Також у каталозі з QOnlineTranslator знаходиться файл README.md, вміст якого відображається під час переходу на сторінку проєкту. Файл містить опис призначення бібліотеки та посилання на документацію, де наведено приклади використання і детально описано всі функції бібліотеки.

Для отримання перекладу потрібно створити об'єкт класу QOnlineTranslator, конструктор якого має такий вигляд:

```
QOnlineTranslator(const QString &text, QString &translationLanguage = "auto", QString &sourceLanguage = "auto", QString &translatorLanguage = "auto").
```

Він конструює об'єкт із перекладом рядка *text*, на мову *translationLanguage* з мови *sourceLanguage* та примітками мовою *translatorLanguage*. Якщо не буде вказано останні параметри під час оголошення об'єкта, то використовуються параметри за замовчуванням.

Конструктор викликає функцію *translate* з такими ж параметрами, яка генерує API-посилання та виконує запит для отримання інформації у форматі JSON. Цей метод описано окремо, оскільки він має важливе самостійне значення. Він розбирає заповнене поле з текстом та повертає його переклад, транслітерацію й інші додаткові аспекти перекладу. Специфіка його опису надає можливість використовувати метод *translate* для отримання іншого варіанту перекладу одного й того ж об'єкта або різних об'єктів безліч разів. А оскільки функція *translate* має специфікатор доступу *public*, то її можна викликати без обмежень доступу у будь-який момент.

Крім функції *translate*, яка є більш загальною, бібліотека містить функції для виконання часткових операцій. Наприклад, за допомогою функції *text()* можна отримати рядок *QString* з перекладом. За допомогою функцій *sourceTranslit()* і *translationTranslit()* відповідно – транслітерацію вхідного та вихідного текстів, за допомогою функцій *sourceLanguage()* і *translationLanguage()* – мову перекладу для вхідного і вихідного текстів. Для отримання різних варіантів перекладу є функція *optionsList()*, яка повертає *QList<QDictionary>*, де *QList* – стандартний контейнер списків у Qt, який містить параметр *QDictionary*, де зберігаються варіанти перекладу. Проілюструємо реалізацію цієї функції.

```
foreach (auto optionType, translator.dictionaryList()) {
    out << optionType.typeOfSpeech() << ":" << endl;
    // Output type of speech with a colon
    for (auto i = 0; i < optionType.count(); i++) {
        if (!optionType.gender(i).isEmpty())
            out << " " << optionType.gender(i); // Prints word gender
            out << " " << optionType.word(i) << ": "; // Prints word
            out << optionType.translations(i); // Prints translations
            out << endl;
        }
    }
    out << endl;
```

}

Наведений фрагмент коду виконує виведення в консоль варіантів перекладу англійського слова «say» на німецьку мову, включаючи список синонімів англійських слів до кожного перекладу німецького слова:

```
"verb:"
"sagen: say, tell, speak, mean, utter"
"sprechen: speak, talk, say, pronounce, militate, discourse"
"meinen: think, mean, believe, say, opine, fancy"
"heißen: mean, be called, be named, bid, tell, be titled"
"äußern: express, comment, speak, voice, say, utter"
"aussprechen: express, pronounce, say, speak, voice, enunciate"
"vorbringen: make, put forward, raise, say, put, bring forward"
"aufsagen: recite, say, speak"
"noun:"
"Sagen: say"
"Mitspracherecht: say"
```

У бібліотеці також наявні статичні функції для визначення та зберігання відомостей про мови перекладу:

`QString codeToLanguage(const QString &code)` – повертає рядок `QString`, що містить код мови для вхідного параметру `code`.

`QString languageToCode(const QString &language)` – повертає рядок `QString`, що містить код для вхідного параметру `language`.

`QString defaultLocaleToCode()` – повертає рядок `QString`, що містить код мови за замовчуванням.

Для випадків, коли слід тільки перекласти або озвучити без потреби створення додаткового об'єкта, є такі статичні функції:

`QString translateText(const QString &text, QString translationLanguage = "auto", QString sourceLanguage = "auto")` – повертає рядок `QString`, який містить переклад рядка `text` з мови `sourceLanguage` на мову `translationLanguage`;

`void say(const QString &text, QString language = "auto")` – озвучує рядок `text` мовою коду вхідного параметра `language`.

Таким чином створена бібліотека дозволяє зручно працювати з перекладом та використовувати її у будь-якому іншому проєкті, що створюється на базі фреймворку Qt.

Із використання описаної бібліотеки розроблено програмний додаток Crow Translate (англ. слова *crow* – ворона і *translate* – перекладати). Сирцевий код пропонованого додатка доступний на сторінці GitHub [21] під ліцензією GPL v3. Для розширення функціональних можливостей проєкту реалізовано два інтерфейси взаємодії – консольний і графічний (за підходом Дуга Макілроя: «Пишіть програми, які б підтримували текстові потоки, оскільки це універсальний інтерфейс»).

Як приклад реалізації консольного інтерфейсу використано інтерфейс програми Translate Shell [16], написаної мовою AWK. Для швидкого доступу з терміналу назву виконавчого файлу скорочено до «crow». Для перекладу тексту в термінальному рядку слід передати виконавчому файлу програми аргумент із текстом для перекладу. Програмою Crow Translate автоматично визначається мова отриманого тексту (використовуючи

Google). Переклад виконується за замовчуванням на мову системи або на мову, яка попередньо задана в налаштуваннях. Наприклад, команда *crow "test words"* виведе у термінал таке:

```
test words
[ English -> Russian ]
тестовые слова
/testovyye slova/
```

Однією з додаткових функцій програмного проєкту є можливість в явному вигляді задати мову вхідного тексту параметром *-s* або *--source* та мову перекладу – параметром *-t* або *--translation*.

Також реалізовано можливість задати мову перекладача параметром *-l* з назвою потрібної мови. Для перекладу відразу на декілька мов потрібно записати назви мов для перекладу зі знаком «+». За умови перекладу тільки одного слова виводяться різні варіанти перекладу.

Наприклад, щоб перекласти слово «*hello*» з англійської мови на українську та німецьку (з російською мовою перекладача за замовчуванням) потрібно виконати таку команду:

```
crow -s en -t uk+de -l ru Hello
У терміналі отримаємо текст:
Hello
[ English -> Ukrainian ]
Здравствуйте
/Zdravstvuyte/
вигук
Алло!: Hello!, Hallo!, Halloa!, Hullo!, Hulloa!
Ало!: Hello!
Вітаю!: Congratulations!, Hello!
Привіт!: Greetings!, Hello!, Hi!, Hallo!, Ave!, Chin-Chin!
[ English -> German ]
Hallo
Ausruf
Hallo!: Hello!, Hi!, Hey!, Howdy!, Yoo-Hoo!
Hallo?: Hello!
Guten Tag!: Hello!, Good day!, Good afternoon!, How do you do?
Servus!: Hello!, Goodbye!, Cheerio!, So long!
```

Текст перекладу можна озвучити за допомогою опції *-e* або *--speak-translation*, а текст оригіналу за допомогою параметру *-q* або *--speak-source*. Для того, щоб тільки озвучити текст, але не виводити переклад, є опція *-a* або *--audio-only*, у процесі виконання якої виводиться тільки текст, що озвучується. Наприклад, наступна команда *crow -t de -a -e "hello"* виводить у термінал переклад на німецьку мову англійського слова «*hello*» та озвучує його: *Hallo*

Для отримання довідки про всі консольні команди потрібно передати опцію *-h* або *--help* на Linux або *-?* – на Windows. Для виведення версії перекладача передбачено опцію *-v* або *-version*. Програма має можливість зчитувати текст з файлів та зі вхідного потоку *stdin*



командами `--file (-f)` та `--stdin (-s)` відповідно.

Під час запуску виконавчого файлу без аргументів відкривається графічний інтерфейс. Під час розробки графічного інтерфейсу вирішено поєднати переваги різних клієнтів для отримання найбільш дружнього інтерфейсу користувача. Основне вікно програми має інтерфейс, схожий на Google Translate, оскільки він звичний для більшості користувачів і в ньому зручно організовано вибір мов (рис. 1). На нижній панелі розташовано кнопки озвучування, швидкого копіювання тексту і меню налаштування програми.

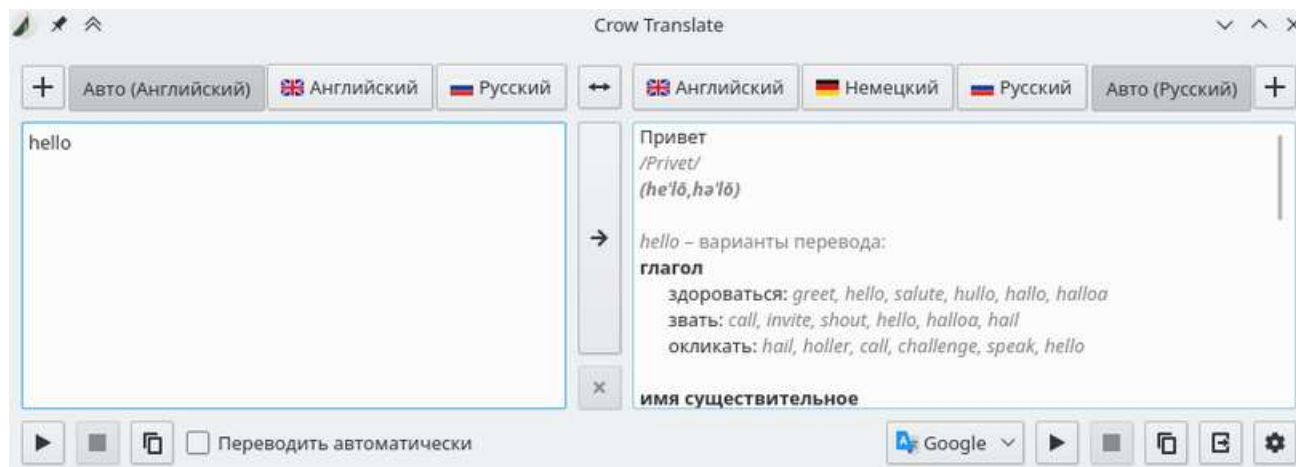


Рис. 1. Головне вікно Crow Translate

У ході аналізу наявних онлайн-перекладачів виділено проблему використання додаткових програм для отримання перекладу. Хоча іноді користувачеві зручніше отримати варіанти перекладу у вікні, що спливає без необхідності додаткового переходу в інші системи. Тому, крім стандартного використання вікна перекладу, розроблену програму можна використовувати як «just-in-time» перекладач виділеного тексту. Для цього достатньо виділити текст і натиснути комбінацію клавіш для перекладу (за замовчуванням це `Ctrl+Alt+E`). Переклад може здійснюватися як в основному, так і у вікнах, що спливають (рис. 2). Тип вікна обирається в налаштуваннях. Для реалізації такої можливості програма має весь час працювати у фоновому режимі. Тому вона згортається в системний трей для зручності, а при перекладі тексту вікно перекладача з'являється знову. Головне вікно можна окремо викликати для введення в нього тексту (за замовчуванням `Ctrl+Alt+C`). Виділений текст можна озвучувати за допомогою гарячих клавіш (за замовчуванням `Ctrl+Alt+S`).

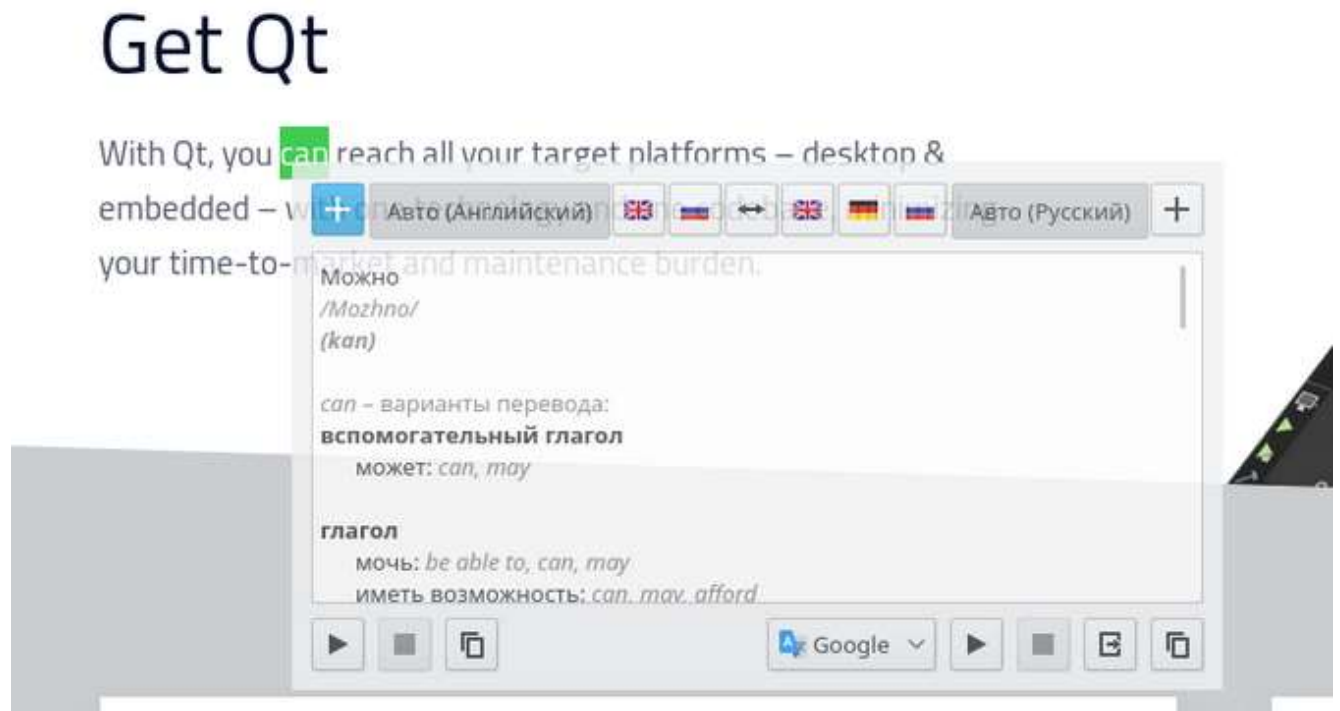


Рис. 2. Вікно Crow Translate, що спливає при перекладі виділеного тексту

У програмі є можливість вибору первинної і вторинної мов для перекладу (за аналогом QTranslate). Наприклад, якщо задати російську мову як первинну, а англійську – як вторинну, то програма за замовчуванням перекладатиме текст на російську, а якщо текст уже є російською мовою, то переклад відбудеться на англійську.

Для зручності використання можна задати мову інтерфейсу програми (рис. 3). Локалізація Crow Translate реалізована за допомогою стандартних засобів Qt, описаних у документації [22]. Завдяки цьому програму можна використовувати будь-якою мовою.

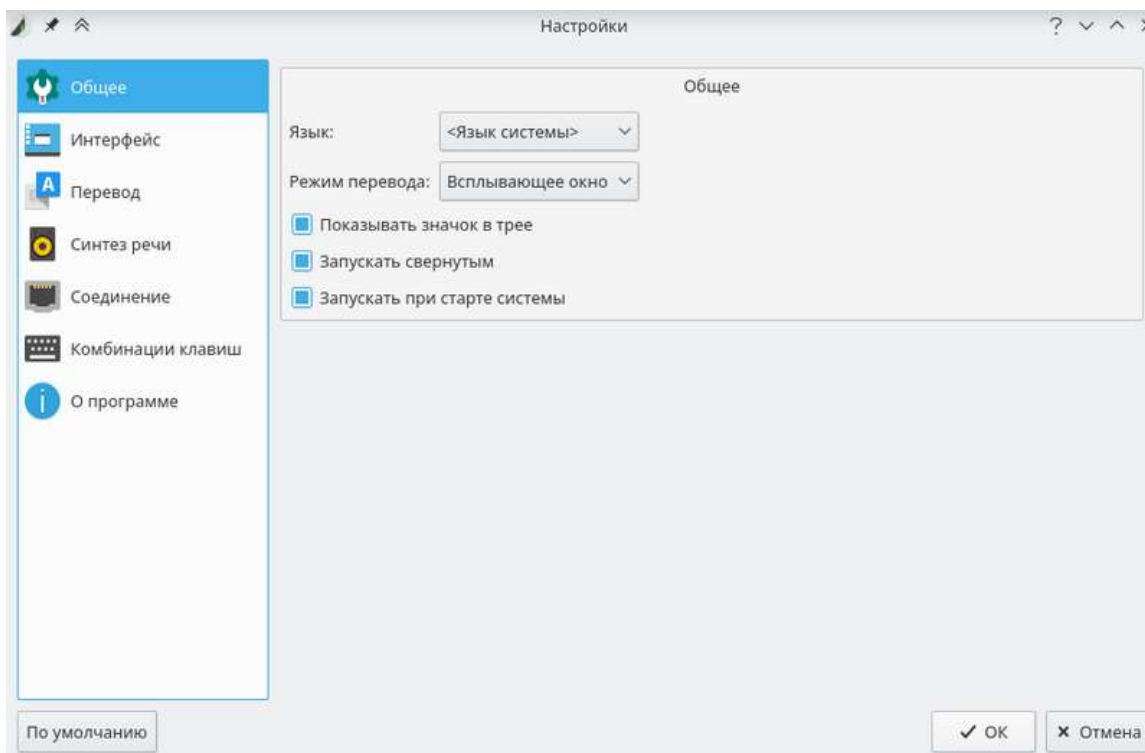


Рис. 3. Меню налаштування Crow Translate

У програмі передбачені налаштування для проксі сервера, вибір типу іконок, вибір прозорості для вікна, що спливає, тощо. Такі можливості додають програмному продукту Crow Translate гнучкості та сприяють зручності використання.

**Результати розробки кросплатформної онлайн-системи для перекладу тексту з відкритим сирцевим кодом.** Завдяки фреймворку Qt інтерфейс програми є нативним і адаптується до будь-якої з операційних систем (Linux, Windows тощо). Але якщо іконки програми для Linux беруться з системної теми системи іконок, то у Windows такого механізму не передбачено. Тому прийнято рішення для платформи Windows використовувати вільні іконки Papirus [23] як значки за замовчуванням, які надають сучасні іконки в стилі Material design.

Розроблена програма Crow Translate споживає мало оперативної пам'яті (приблизно 19 Мб) і майже не навантажує малопотужні процесори, але при цьому має багато налаштувань, що робить програмний продукт гнучким та затребуваним.

Зазначимо можливості поширення програми. Для дистрибутивів Linux на базі Debian, таких, як Ubuntu, Linux Mint, Kali Linux, Tails, Elementary OS був підготовлений deb-пакет. Його можна завантажити і встановити зі сторінки проекту на GitHub [21] в меню «Releases». Для користувачів дистрибутивів на базі Arch Linux, таких як, Manjaro, Chakra Linux, Antergos тощо, підготовлено PKGBUILD і додано в AUR під назвою crow-translate-git [24], який можна встановити, використовуючи пакетний менеджер або зробити це вручну.

У репозиторії з програмою знаходиться скрипт для Linux, який автоматично створює

дистрибутив і збирає під нього пакет з останніми змінами. Зазначене вище, а також інструкції для ручної компіляції задокументовані в файлі README.md на GitHub [21].

Для платформи Windows підготовлений інсталятор (завантажити який можна на вкладці «Releases») для встановлення програми, створення значків у меню «Пуск» і запису інформації про встановлення у реєстр. Програма відображається у меню «Встановлення і видалення програм», де також доступний деінсталятор. Інсталятор створено за допомогою Nullsoft Scriptable Install System (NSIS) – система генерації інсталяторів програм для Microsoft Windows із відкритими сирцевими кодами [25]. Система описує скриптову мову, на основі якої написано код інсталятора, який збирається за допомогою компілятора в звичайний виконуваний файл. NSIS також використовувалася такими популярними програмами, як Антивірус Касперського, Notepad++, VLC Player, QBitTorrent, OpenVPN тощо, оскільки вона відкрита, функціональна і має докладну документацію. Отже, описана система дозволяє створювати сучасний і зручний для користувача інсталятор на основі вільних технологій.

**Обговорення результатів розробки кросплатформної онлайн-системи для перекладу тексту з відкритим сирцевим кодом.** Необхідність опису розробки кросплатформних систем на базі сучасного фреймворку пояснюється відсутністю наукових напрацювань практичного спрямування, які б робили акцент саме на способах та технологіях створення таких систем. Процес розроблення продемонстровано на прикладі програмної системи для перекладу тексту з відкритим сирцевим кодом. Пропонована наукова публікація акцентує увагу на програмній реалізації онлайн-перекладача тексту в обраній системі та особливостях розробки власного програмного продукту CrowTranslate.

У процесі дослідження продемонстровано етапи розробки програмного забезпечення. Першим етапом став аналіз та виявлення наявних розробок із зазначеної проблематики для дослідження особливостей аналогічних програмних систем перекладу текстів. У запропонованій системі Crow Translate враховані виявлені недоліки. Для роботи Crow Translate не треба окремо завантажувати словники. Вона працює у фоновому режимі та не потребує відкриття браузера для отримання перекладу або додаткового переходу в інші системи.

У межах цього дослідження неможливо виконати докладний огляд усіх мов програмування та фреймворків. Однією з причин цього є те, що не існує чітких критеріїв вибору залежно від поставлених задач. Програміст виконує такий вибір тільки за власними уподобаннями та з урахуванням вимог замовника до функціональних можливостей проєкту. Так само не можна охопити і проаналізувати всі наявні аналоги систем-перекладачів. Як правило, у реальній професійній діяльності програмне забезпечення такого типу розробляється на замовлення та за вимогами замовника і не завжди є абсолютно унікальним програмним продуктом.

Розвиток запропонованого дослідження можливий за декількома аспектами. Подальшого вдосконалення потребують питання детального опису етапу проєктування з побудовою функціональних та структурних діаграм. Також перспективним у подальшому вдосконаленні програмної системи є покращення алгоритму перекладу, підтримка додаткових онлайн-сервісів із можливістю швидкого перемикання між ними.

**Висновки.** 1. Проаналізовано такі сервіси здійснення перекладу, як Google Translate, Bing Translator, DeepL Translator і Apertium. Як прототип обрано Google Translate з огляду

на його популярність, великий набір кількості мов і якість перекладу. Установлено, що з доступних для перегляду та дослідження онлайн-перекладачів для Linux не виявлено жодного сучасного перекладача з графічним інтерфейсом, а для Windows – жодного з відкритим сирцевим кодом. Деякі з оглянутих систем є комерційними проектами або мають лімітовані функції у безкоштовній версії, що значно обмежує їх аудиторію.

2. Досліджено переваги використання мови C++ та обґрунтовано вибір кросплатформного фреймворку Qt серед інших аналогів. Проведений огляд сучасних мов програмування та фреймворків для створення програмних додатків дозволив обрати сучасний фреймворк Qt для швидкої розробки графічного інтерфейсу як такий, що задовольняє вимогам до розроблення власного програмного продукту.

3. Під час реалізації програми-перекладача прийнято рішення розділити процес перекладу та інтерфейс програми на два окремих проєкта. Така модульність спростила супровід коду і дозволила використовувати механізм перекладу як окрему бібліотеку в інших проєктах. Додатковою особливістю розробленого проєкту стала окрема бібліотека QOnlineTranslator для перекладу тексту. Вона є вільним програмним забезпеченням та опублікована на сторінці GitHub під ліцензією GPL v3 для використання у будь-яких аналогічних проєктах.

4. Надано докладний опис технології створення повнофункціонального кросплатформного онлайн-перекладача Crow Translate з відкритим сирцевим кодом. Останній забезпечує отримання перекладу та озвучування виділеного тексту, має підтримку роботи через консольний та графічний інтерфейси, гнучке налаштування і підтримку гарячих клавіш. Висока якість перекладу забезпечується завдяки використанню Google API. Є можливість роботи в режимі словника під час введення одного слова, відображаючи варіанти перекладу на обрану мову та у транслітерації. Програмний продукт має зручний доступ до перекладу виділеного тексту в будь-якому додатку та потребує невеликого споживання оперативної пам'яті (до 19 Мб).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Арестова, А. А.. Сравнительный анализ систем машинного перевода. Вестник ВолГУ. 2015, Вып. 13. С. 105–109.
2. Горелик, П. Л. Использование ресурсов сети Интернет при переводе медицинского текста (на материале английского языка). Гуманитарные, социально-экономические и общественные науки. 2014. № 7. С. 295–298.
3. Денисова Д. С. Современные системы машинного перевода. Статический машинный перевод. Синергия наук. 2018. № 19. С. 1425–1434.
4. Максименко О. И., Чинина, Д. С. Обзор системы машинного перевода «Google-переводчик» (на примере финского языка). Science Time. 2014. № 5. С. 133–139.
5. Назарчук Ю. И Сравнительная характеристика автоматизированных систем CAT и TRADOS, используемых в переводческой деятельности. Современные лингвистические и методико-дидактические исследования. 2017. № (34). С. 180–189.
6. Усов С. С. Использование современных технологий в изучении лексики английского языка. Синергия наук. 2018. № 19. С. 1387–1392.
7. Кадникова О. В., Крамаренко, В. А. Преимущества и недостатки машинного перевода на примере популярных on-line переводчиков. Филологический аспект.

2017. № 6 (26). С. 86–96.
8. Banik D., Ekbal A., Pushpak, A., Statistical machine translation based on weighted syntax-semantics, Sadhana. Academy Proceedings in Engineering Sciences. 2020. Volume 45. Issue 1. P. 191.
  9. Abarghooei M. Designing a Cross-Platform Mobile Learning System. Lecture Notes on Software Engineering. 2015. Vol. 3. No. 3. P. 195–198.
  10. Ronglong S., Arpnikanondt, C. An open-source cross-platform universal messaging system with feedback support. Journal of Systems and Software. 2016. Volume 117. P. 30–54.
  11. Heitkötter H., Hanschke S., Majchrzak, T. A. Evaluating cross-platform development approaches for mobile applications. Lecture Notes in Business Information Processing. 2013. Volume 140. P. 120–138.
  12. Microsoft Translator. Microsoft. URL: <https://translator.microsoft.com/ru/apps/>
  13. Client for Google Translate. TranslateClient. URL: <http://translateclient.com/>
  14. QTranslate. QuestSoft. URL: <https://quest-app.appspot.com>
  15. Компьютерный перевод от Trident Software. Pragma. Trident Software. URL: <http://www.translate.ua>
  16. Translate Shell. Soimort. URL: <https://www.soimort.org/translate-shell/>
  17. Most Popular Technologies: Programming, Scripting, and Markup Languages. Stack Overflow. URL: <https://insights.stackoverflow.com/survey/2018/#technology-programming-scripting-and-markup-languages>
  18. Octoverse. GitHub. URL: <https://octoverse.github.com/>
  19. QOnlineTranslator. Shatur95. URL: <https://github.com/Shatur95/QOnlineTranslator>
  20. GPL v3. GNU. URL: <https://www.gnu.org/licenses/gpl-3.0.en.html>
  21. Crow Translate. GitHub. URL: <https://github.com/Shatur95/Crow-Translate>
  22. Internationalization with Qt. Qt Documentation. URL: <https://doc.qt.io/qt-5/internationalization.html>
  23. Papyrus. GitHub. URL: <https://github.com/PapyrusDevelopmentTeam/papyrus-icon-theme>
  24. Crow-translate-git. AUR. URL: <https://aur.archlinux.org/packages/crow-translate-git/>
  25. NSIS. SourceForge. URL: <http://nsis.sourceforge.net>

*Матеріал надіслано до редакції 20.04.2021р.*

## **THE CONSTRUCTION METHODS FOR SOFTWARE COMPLEXES BY THE EXAMPLE OF AN OPEN CROSS-PLATFORM ONLINE SYSTEM FOR TEXT TRANSLATION**

### **Hanna Pohromska**

PhD in Pedagogics, Docent, Assistant professor of the department of theory and methods for teaching the natural sciences, mathematics and information technologies

Mykolaiv in Service Teacher Training Institute, Mykolaiv, Ukraine

*hanna.pohromska@moippo.mk.ua*

ORCID: 0000-0002-6779-3995

**Natalya Makhrovskaya**

PhD in Mathematics and Statistics, Assistant professor of the department of theory and methods for teaching the natural sciences, mathematics and information technologies

Mykolaiv in Service Teacher Training Institute, Mykolaiv, Ukraine

*natalya.makhrovskaya@moippo.mk.ua*

ORCID: 0000-0001-9603-6902

**Abstract.** Popular online translators Google Translate, Bing Translator, DeepL Translator, and Apertium are reviewed. The goal is to compare the interfaces and functionality of these translators. The advantages of using the C++ programming language are investigated. The choice of the Qt cross-platform framework among other analogues has been substantiated. Qt most of all meets the requirements set in the article for the development of its own software product. The basis for the development of the program is chosen methods of object-oriented programming using the application programming interface. The stages of development of the online «Crow Translate» system are presented. Its capabilities for text translation are described: an application programming interface for translation, peculiarities of the implementation of a C++ programming system based on the Qt framework, basic commands of the console interface, functionality of the graphical interface for dubbing and translation, customization options, localization possibilities and ways of distributing the offered «Crow Translate» software. During the development of the system, the implementation of the translation process and the program interface were divided into two separate projects. A peculiarity of the project implementation is the development of a separate QOnlineTranslator library for text translation. This system is free software and is published on the GitHub page under the GPL v3 license for use in any similar projects. The details of the implementation of the suggested library and its main functions are described. Thanks to the use of the Google interface in the described program, a high-quality translation is ensured. The developed project provides for the ability to work in the dictionary mode when entering one word or a piece of text. Various translation options are additionally displayed for one word. There is support for displaying transliteration. When developing a software product, the possibility of receiving a translation in a pop-up window without additional transitions to the browser or other applications was taken into account. The result is an open cross-platform software product that has convenient access to the translation of a text selected in any application and takes up a small amount of RAM.

**Keywords:** online translators; C++ cross-platform system development; Qt; Google API; source code

**REFERENCES (TRANSLATED AND TRANSLITERATED)**

1. Arestova, A. (2015). Comparative analysis of machine translation systems. Bulletin of VolsU. series 9, issue. 13. (in Russian)
2. Gorelik, P. (2014). The use of Internet resources in the translation of a medical text (based on English). Humanitarian, socio-economic and social sciences. volume 7. (in Russian)
3. Denisova, D. (2018). Modern systems of machine translation. Static machine translation. Synergy of Sciences, volume. 19. (in Russian)
4. Maksimenko, O. & Chinina, D. (2014). Review of the machine translation system “Google-translator” (on the example of the Finnish language). Science Time, volume 5. (in Russian)
5. Nazarchuk, Yu. (2017). Comparative characteristics of automated systems CAT and TRADOS used in translation activities. Modern Linguistic and Methodological-Didactic Studies, volume 2 (34). (in Russian)
6. Usov, S. (2018). The use of modern technology in the study of vocabulary of the English language. Synergy of Science, volume 19. (in Russian)
7. Kadnikova, O. & Kramarenko, V. (2017). Advantages and disadvantages of machine

- translation on the example of popular online translators. Philological aspect, volume 6 (26). (in Russian)
8. Banik D., & Ekbal A., & Pushpak, A. (2020). Statistical machine translation based on weighted syntax–semantics, Sadhana. Academy Proceedings in Engineering Sciences. Volume 45. Issue 1.
  9. Abarghooei, M. (2015). Designing a Cross-Platform Mobile Learning System. Lecture Notes on Software Engineering, volume 3.
  10. Ronglong, S. & Arpnikanondt, C. (2016). An open-source cross-platform universal messaging system with feedback support. Journal of Systems and Software. volume 117.
  11. Heitkötter, H., & Hanschke, S., & Majchrzak, T. A. (2013) Evaluating cross-platform development approaches for mobile applications. Lecture Notes in Business Information Processing. International Conference on Web Information Systems and Technologies, volume 140.
  12. Microsoft Translator. (2018). November 27, 2018. <https://translator.microsoft.com/ru/apps/>
  13. Client for Google Translate. (2018). TranslateClient. December 12, 2018. <http://translateclient.com/>
  14. QTranslate. (2019). QuestSoft. January 11, 2019. <https://quest-app.appspot.com>
  15. Computer Translation from Trident Software. Pragma. (2018). Trident Software. December 12, 2018. <http://www.translate.ua> (in Russian)
  16. Translate Shell. (2020). Soimor. December 20, 2020. <https://www.soimort.org/translate-shell/>
  17. Most Popular Technologies: Programming, Scripting, and Markup Languages (2018). Stack Overflow. December 7, 2018. <https://insights.stackoverflow.com/survey/2018/#technology-programming-scripting-and-markup-languages>
  18. Octoverse. (2018). GitHub. November 21, 2018. <https://octoverse.github.com/>
  19. QOnlineTranslator. (2019). Shatur95. October 19, 2019. <https://github.com/Shatur95/QOnlineTranslator>
  20. GPL v3. (2018). GNU. November 10, 2018. <https://www.gnu.org/licenses/gpl-3.0.en.html>
  21. Crow Translate. (2018). GitHub. <https://github.com/Shatur95/Crow-Translate>
  22. Internationalization with Qt. Qt Documentation. November 8, 2018. <https://doc.qt.io/qt-5/internationalization.html>
  23. Papyrus. (2020). GitHub. November 8, 2020. <https://github.com/PapyrusDevelopmentTeam/papyrus-icon-theme>
  24. Crow-translate-git. (2018). AUR. November 17, 2018. <https://aur.archlinux.org/packages/crow-translate-git/>
  25. NSIS. (2020). SourceForge. October 7, 2020. <http://nsis.sourceforge.net>