

ПРАБЛЕМЫ ЛАКАЛІЗАЦЫІ ПРАГРАМНАГА ЗАБЕСПЯЧЭННЯ НА БЕЛАРУСКУЮ МОВУ

Лагуцінава В.А.

*Беларускі дзяржаўны ўніверсітэт інфарматыкі і радыёэлектронікі
г. Мінск, Рэспубліка Беларусь*

Шматкова І.І. – канд. філал. навук

Артывул апісвае метады лакалізацыі Python на беларускую мову з выкарыстаннем **GNUgettext** і **Babel** для перакладу тэксту, фарматавання дат і праблем Unicode. Распрацаваны ІТ-гласарый, вызначаны праблемы (сінанімія, адсутнасць літар у шрыфтах) і прапанавана іх вырашэнне. Лакалізацыя разглядаецца як культурная ініцыятыва для захавання мовы ў ІТ-сферы праз удзел навуковай супольнасці, супрацоўніцтва з універсітэтамі і міжнароднымі платформамі.

Лакалізацыя праграмнага забеспячэння – ключавы этап для захавання і папулярызацыі беларускай мовы ў ІТ-сферы. Згодна з *Ethnologue*, беларуская мова адносіцца да «уразлівых», што робіць яе падтрымку ў тэхналогіях задачай нацыянальнай важнасці [1]. Нягледзячы на пазіцыі

Python у рэйтынгі TIOBE (3-е месца, 2025), поўная лакалізацыя гэтай мовы праграмавання адсутнічае, што абмяжоўвае яе выкарыстанне ў адукацыі і стартапах.

Мэта працы: распрацаваць падыход да лакалізацыі Python на беларускую мову, уключаючы інтэрфейс, дакументацыю і сістэмныя паведамленні.

Задачы:

- аналіз інструментаў лакалізацыі (GNUgettext, Babel);
- стварэнне ІТ-гласарыя на беларускай мове;
- рэалізацыя перакладу кампанентаў Python;
- выяўленне праблем (тэрміналогія, Unicode) і прапанова іх вырашэння.

Тэарэтычныя асновы: інтэрналізацыя і лакалізацыя.

Інтэрналізацыя – падрыхтоўка кода для падтрымкі шматмоўя (напр., фарматы дат, валют).

Лакалізацыя – адаптацыя прадукту пад мову, культурныя саблівасці і рэгіянальныя налады (напр., пераклад тэкстаў, правакшыфтоў). У кантэксце Python інтэрналізацыя ажыццяўляецца праз стандартныя бібліятэкі (gettext), а лакалізацыя патрабуе стварэння моўных рэсурсаў.

Стандартным інструментам для лакалізацыі ў Python з'яўляецца GNU gettext. Ён дазваляе перакладаць радкі ў кодзе, ствараць файлы перакладаў (накштат .po) і кіраваць лакалізацыяй.

Прынцып працы на два працузвычайнага перакладчыка: узяты тэкст вырадок зыходнага кода перадаецца ў тыліце «pygettext» або «xgettext», якая вылучае яго ў шаблон (.pot). Перакладчык запаўняе гэты шаблон перакладамі, захоўваючы структуру ў .po-файле. Затым гэты файл кампілюецца ў бінарны .mo-файл, які ўжываецца ў праграме. Для беларускай мовы пры гэтым выкарыстоўваецца код лакалізацыі_BY [3],[2].

Альтэрнатывым спосабам можа стаць выкарыстанне бібліятэкі Babel. Яна не толькі пашырае функцыянальнасць «gettext», але і вырашае праблемы, звязаныя з культурнымі саблівасцямі. Напрыклад, фарматаванне дат на беларускай мове патрабуе дакладнага ўліку склонаў («5-га сакавіка» замест проста «5 сакавіка»). Аднак застаецца праблема з адсутнасцю некаторых літар у стандартызаваным Unicode UTF-8 (напрыклад, літара «ў» часта адлюстроўваецца няправільна ў шрыфтах, што ўскладняе поўнаўвартасны пераклад) [4].

Першы з магчымых прыкладаў перакладу Python на беларускую мову (табліца 1):

Табліца 1 – Першы з магчымых прыкладаў перакладу Python на беларускую мову

Loop		Цыкл
String		Радок
Debug		Адладка
Syntax		Сінтаксіс
Link		Спасылка
Log		Журнал

Праблемы, якія ўзніклі.

Беларуская мова, дзякуючы сваёй багатай лексіцы, прыяла лакалізацыі мае складанасці з выбарам адзіных тэрмінаў. Вялікая колькасць сінонімаў (напрыклад, «адладка» і «дыягностыка») прыводзіць да неадпаведнасцей пры перакладзе ІТ-тэрміналогіі [6]. Для вырашэння гэтай праблемы прапануецца выкарыстоўваць платформы тыпу GitHub, дзе можна праводзіць адкрытыя дыскусіі і галасаванні за найбольш удалыя варыянты. (Дадаткова: прыклад такога абмеркавання – спрэчка вакол перакладу «debug» як «адладка» ці «адхіленне памылак»).

Праблема з Unicode аказалася больш сур'ёзнай. Нават для напісання гэтага тэксту прыйшлося шукаць шрыфт, які карэктна адлюстроўвае ўсе беларускія літары (напрыклад, «ў», «ё»). Рашэнне: усталяваць спецыяльны шрыфт з падтрымкай беларускай мовы (напрыклад, *Noto Sans*). Стварыць дадатак да існуючых шрыфтоў, які дадае адсутныя сімвалы. (Дадаткова: праблематычней не толькі тэхнічна частка, але і стандартызацыя — не ўсе праграмы аўтаматычна распазнаюць беларускія сімвалы) [5].

Лакалізацыя Python на беларускую мову – гэта, перш за ўсё, культурная ініцыятыва, а не проста тэхнічны праект. Паспяховыя прыклады (*Ubuntu*, *Wikipedia*) паказваюць, што нават мовы з абмежаванай колькасцю носьбітаў могуць існаваць у лічбавай прасторы.

Спіс выкарыстаных крыніц:

1. Ethnologue: Languages of the World [Электронны рэсурс]. – Рэжым доступу: <https://www.ethnologue.com> – Дата доступу: 15.04.2025.

2. Локалізацыя праграм: вопыт Astralinux [Электронны рэсурс]. – Рэжым доступу: <https://habr.com/ru/companies/astralinux/articles/861814/>. – Дата доступу: 15.04.2025.
3. Дакументацыя Python: модуль gettext [Электронны рэсурс]. – Рэжым доступу: <https://docs.python.org/3/library/gettext.html>. – Дата доступу: 15.04.2025
4. Метады шматмоўнасці ў Python [Электронны рэсурс]. – Рэжым доступу: <https://ru.stackoverflow.com/questions/521034/Способы-мультязычности-программ-на-python>. – Дата доступу: 15.04.2025
5. Пераклад тэкставых рэсурсаў [Электронны рэсурс]. – Рэжым доступу: <https://habr.com/ru/companies/otus/articles/523112/>. – Дата доступу: 15.04.2025
6. Беларускі ІТ-слоўнік [Электронны рэсурс]. – Рэжым доступу: <https://belazar.info/belsoft/compdict.php>. – Дата доступу: 15.04.2025