



Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Практична робота №1

Економіка ІТ-індустрії

Тема: Приріст росту популярності мови програмування Rust

Виконали

студенти групи ІІІ-11:

Панченко С. В., Сідак К. І.,

Тихонов Ф. С., Рябов Ю. І.

Перевірів:

Родіонов П.В.

Київ 2024

АНОТАЦІЯ

Практична робота містить 18 сторінок, 9 рисунків, 8 джерел.

Практична робота присвячена аналізу росту популярності мови програмування Rust[1].

Мета: проаналізувати та оцінити динаміку популярності та використання мови програмування Rust протягом останніх років, визначити тенденції активності використання та порівняти з іншими популярними мовами програмування.

ЗМІСТ

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	4
2 ВИКОНАННЯ.....	5
3.1 Пошуки у Google.....	5
3.2 ТЮВЕ та PYPL.....	7
3.4 Аналіз статистики pull-requests, stars, pushes, issues у публічних репозиторіях Github.....	8
ВИСНОВОК.....	13
ПОСИЛАННЯ.....	15
ДОДАТОК А.....	16

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

Rust - це сучасна системна мова програмування, яка набуває все більшої популярності серед розробників завдяки своїй унікальній комбінації безпеки пам'яті, конкурентності та продуктивності. Розроблена Mozilla Research, мова Rust вперше з'явилася у 2010 році і з тих пір стрімко розвивається, знаходячи застосування у різноманітних сферах розробки програмного забезпечення.

Rust використовується для створення операційних систем, браузерних компонентів, ігрових движків, а також у сферах, де критично важливі продуктивність та безпека - від вбудованих систем до великомасштабних веб-сервісів. Його застосовують такі технологічні гіганти, як Mozilla, Microsoft, Google та Amazon, що свідчить про зростаючу довіру до мови в індустрії.

Наше дослідження має особливе значення в контексті стрімкого розвитку технологій та зростаючої потреби в надійному та ефективному програмному забезпеченні. Аналізуючи тренди пошукових запитів, пов'язаних з Rust та його ключовими бібліотеками, ми прагнемо продемонструвати динаміку розвитку мови та інтересу до неї серед розробників.

Кінцевою метою цієї роботи є показати, що Rust є технологією зі стрімким розвитком і значним потенціалом. Ми розглянемо не лише загальний інтерес до мови, але й тенденції у використанні важливих бібліотек, таких як Tokio для асинхронного програмування та Serde для серіалізації даних. Це дозволить нам отримати комплексне уявлення про те, як Rust еволюціонує та адаптується до сучасних потреб розробки програмного забезпечення.

Цей аналіз буде корисним для розробників, які розглядають можливість вивчення нових технологій, для компаній, що планують технологічні стратегії, а також для дослідників, які вивчають тенденції в галузі програмування. Розуміння траєкторії розвитку Rust допоможе прогнозувати його майбутній вплив на індустрію та потенційні сфери застосування.

2 ВИКОНАННЯ

3.1 Пошуки у Google

Google Trends [2] - це безкоштовний онлайн-інструмент та інформаційний сервіс від компанії Google, який дозволяє аналізувати популярність і частоту пошукових запитів різних ключових слів та фраз у пошуковій системі Google.

Рисунок 3.1 відображає загальний інтерес до мови програмування Rust. Графік показує стабільне зростання інтересу з 2015 року, з помітним прискоренням зростання з початку 2021 року. Пік інтересу спостерігається наприкінці 2023 року. Це свідчить про зростаючу популярність Rust як мови програмування.

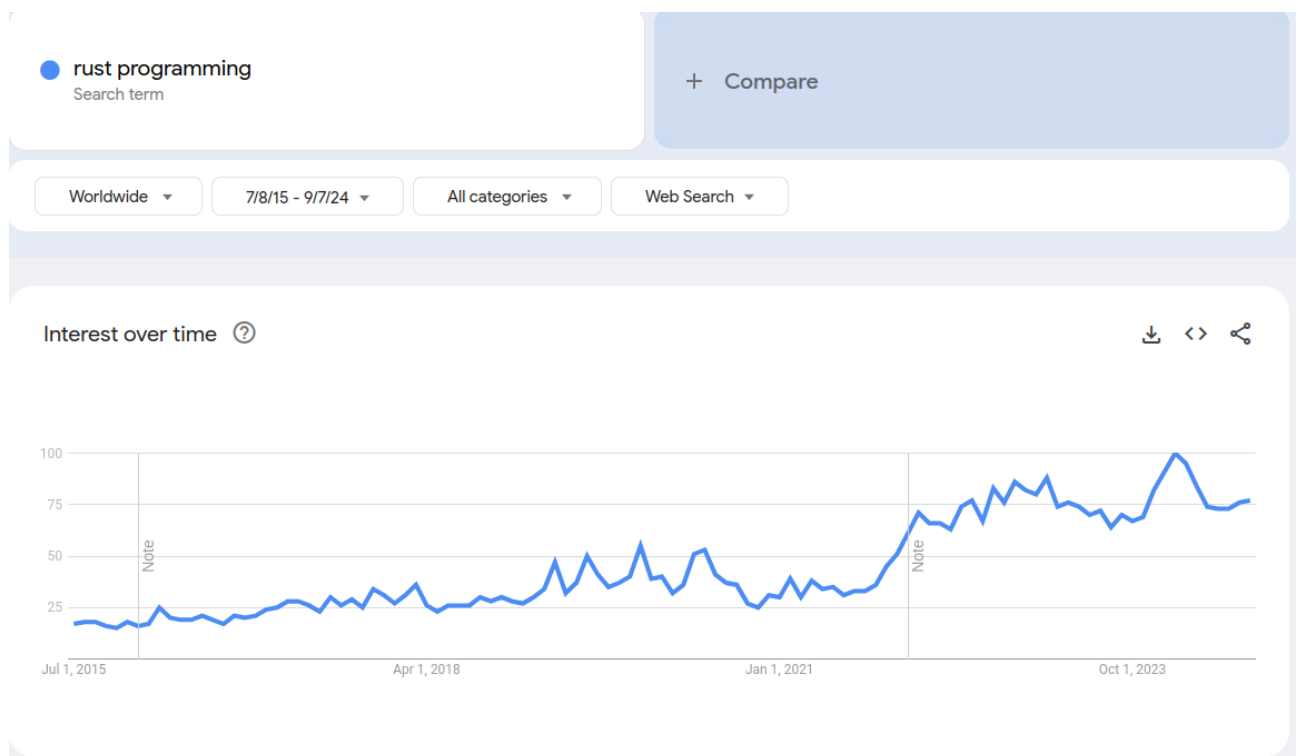


Рисунок 3.1 – Графік відносної кількості пошукових запитів rust programming до пікової кількості за даний період

Tokio[3] - це асинхронний runtime для Rust, який використовується для створення надійних мережевих застосунків. Рисунок 3.2 показує різке зростання інтересу з початку 2021 року, що може бути пов'язано з збільшенням використання Rust для розробки серверних та мережевих застосунків. Вибір цієї бібліотеки для аналізу вказує на зростаючу важливість асинхронного

програмування в екосистемі Rust.

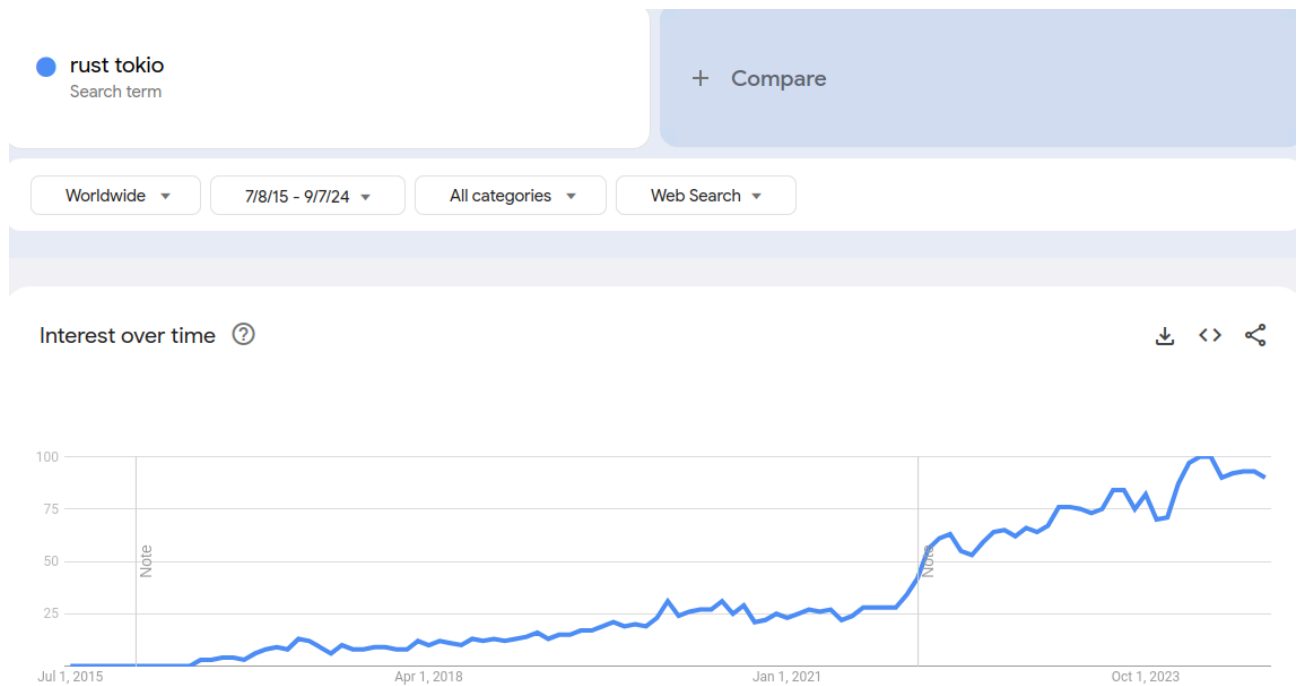


Рисунок 3.2 – Графік відносної кількості пошукових запитів rust tokio до пікової кількості за даний період

Serde[4] - це фреймворк для серіалізації та десеріалізації даних в Rust. Графік демонструє поступове зростання інтересу з 2015 року, з помітним стрибком у жовтні 2021 року (як зазначено на графіку). Після цього спостерігається стабільно високий рівень інтересу. Вибір Serde для аналізу підкреслює важливість обробки даних у проектах на Rust.

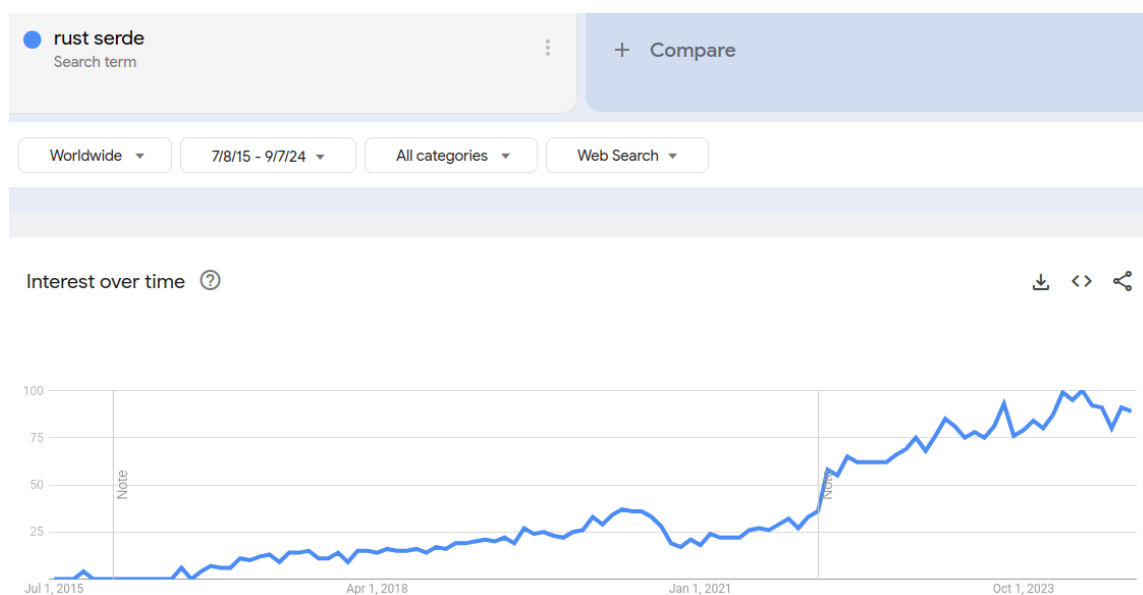


Рисунок 3.3 – Графік відносної кількості пошукових запитів rust serde до пікової кількості за даний період

3.2 TIOBE та PYPL

Однією із популярних метрик оцінювання популярності мови програмування є індекс TIOBE[5], який розраховується на основі кількості результатів у пошукових системах для запитів, що містять назву даної мови програмування. Індекс охоплює пошуки в Google, Google Blogs, MSN, Yahoo!, Baidu, Wikipedia та YouTube, а оновлення індексу відбувається раз на місяць. Графік зміни цього індексу для мови RUST наведений на рисунку 3.4.

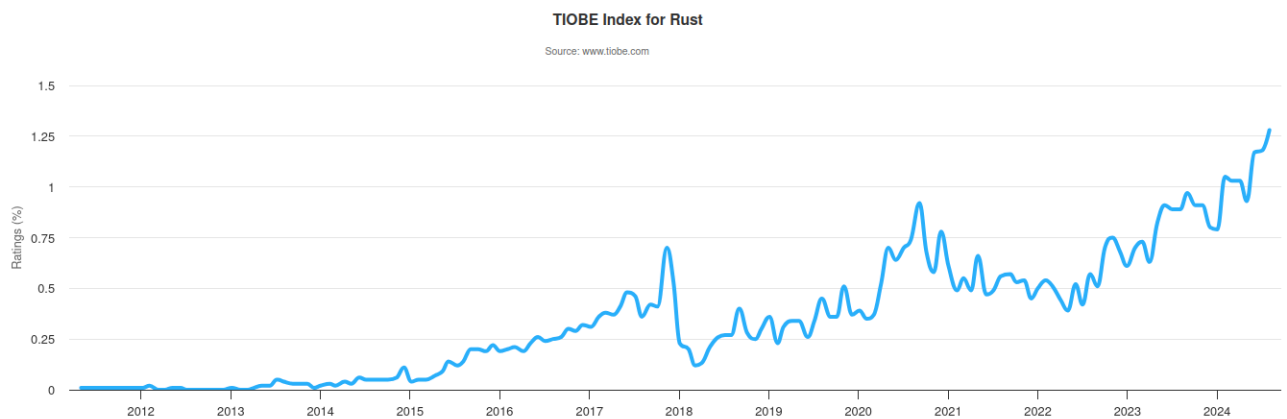


Рисунок 3.4 – Графік зміни індексу TIOBE за даний період

Можна побачити, що Rust має чіткий зростаючий тренд, що може свідчити про збільшення його популярності.

Хоча індекс TIOBE й використовують для оцінювання та порівняння популярності мов програмування, проте він має один суттєвий недолік. Він підраховує саме кількість веб-сторінок з назвою мови, причому кількість пошукових запитів не впливає на цей показник. Наприклад, сторінок в Інтернеті, пов'язаних з Objective-C, понад 20 мільйонів, в той час як про C – лише 11 мільйонів. Це пояснює, чому Objective-C має високий рейтинг TIOBE, проте в реальності відносно небагато людей читають ці веб-сторінки на Objective-C, а сама мова на даний момент є значно менш популярною за C. У той же час програмування на Objective C шукають у 30 разів рідше, ніж програмування на C. Більш об'єктивною метрикою є індекс популярності мови програмування PYPL[6], який показує, як часто шукають tutorіали з певної

мови програмування в Google: чим частіше шукають їх шукають, тим популярнішою вважається ця мова. Вихідні дані беруться з Google Trends. Графік зміни цього індексу для мови RUST наведений на рисунку 3.5.



Рисунок 3.5 – Графік зміни індексу PYPL за даний період

Даний графік також показує позитивну динаміку, що дозволяє впевнено стверджувати про зростаючу популярність мови Rust.

3.4 Аналіз статистики pull-requests, stars, pushes, issues у публічних репозиторіях Github

Madnight.github.io[7] - це сторінка на Github[8], що показує статистику з pull-requests, stars, pushes, issues у публічних репозиторіях, починаючи з 2012 року. Цей ресурс дозволяє завантажити дані зі статистикою, які потім можна проаналізувати.

Після обробки даних ми можемо вивести статистику відношення кількості pull-requests, stars, pushes, issues мови програмування X до пікової кількості pull-requests, stars, pushes, issues відповідно мови програмування X.

Синім кольором показано статистику для Rust.

Розглянемо рисунки 3.6, 3.7, 3.8, 3.9.

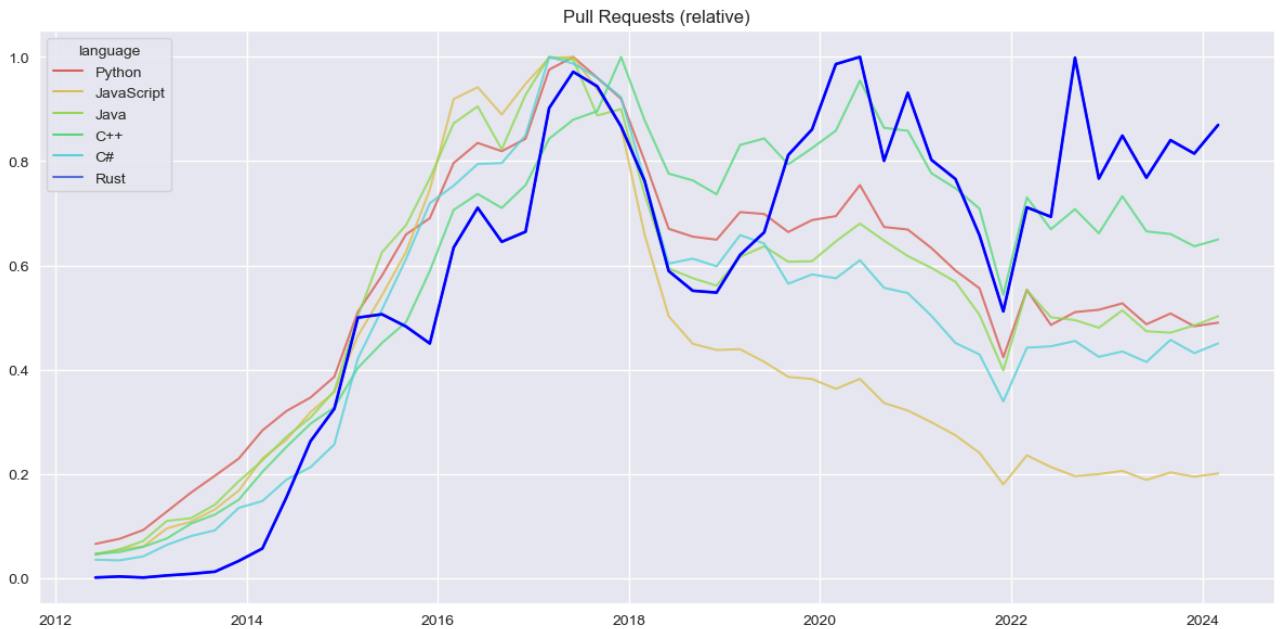


Рисунок 3.6 – Графік відносної кількості пул реквестів у публічних репозиторіях GitHub до пікової кількості для кожної з найбільш популярних мов програмування за даний період

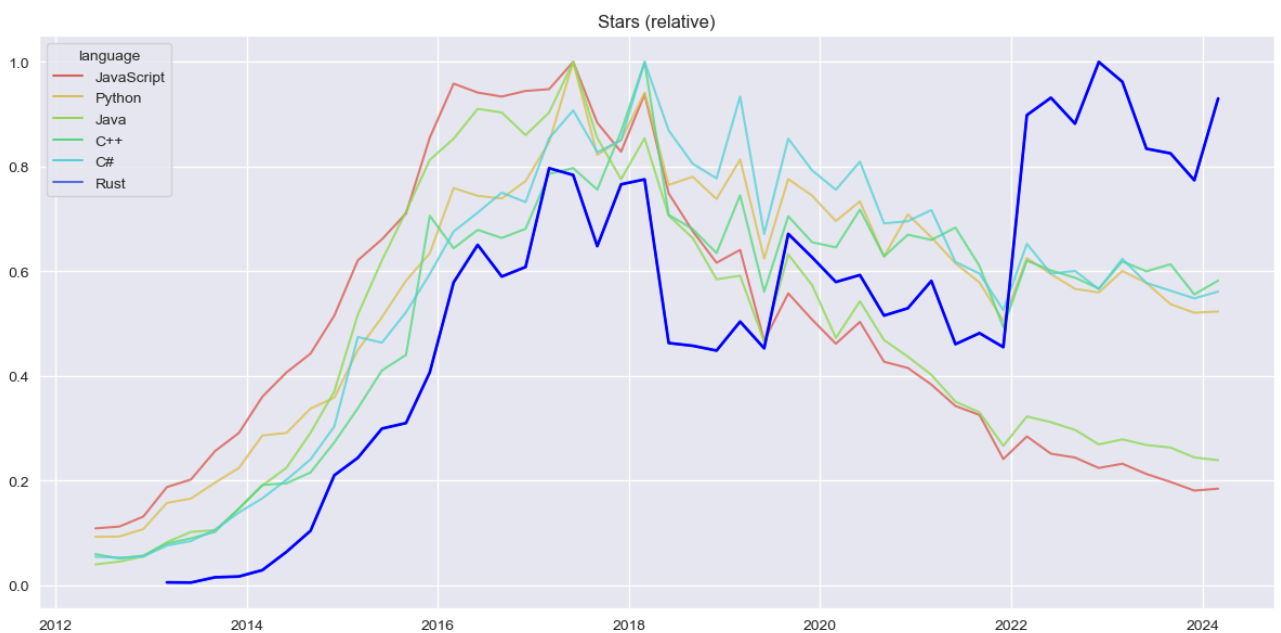


Рисунок 3.7 – Графік відносної кількості stars у публічних репозиторіях GitHub до пікової кількості для кожної з найбільш популярних мов програмування за даний період

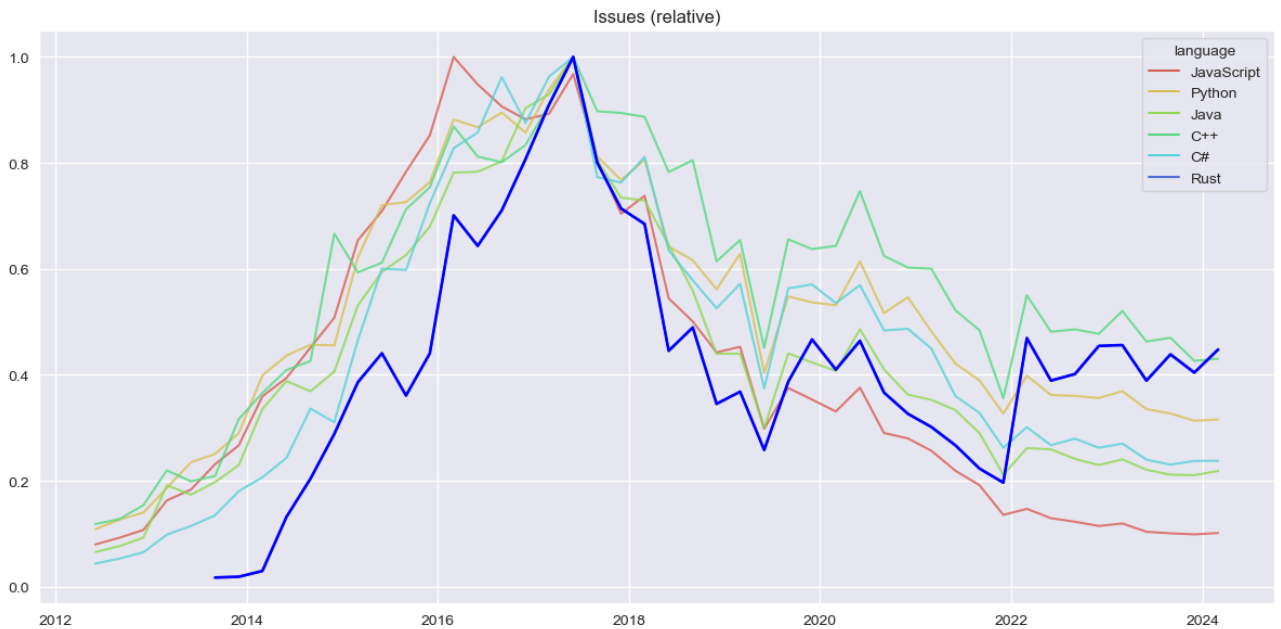


Рисунок 3.8 – Графік відносної кількості issues у публічних репозиторіях GitHub до пікової кількості для кожної з найбільш популярних мов програмування за даний період

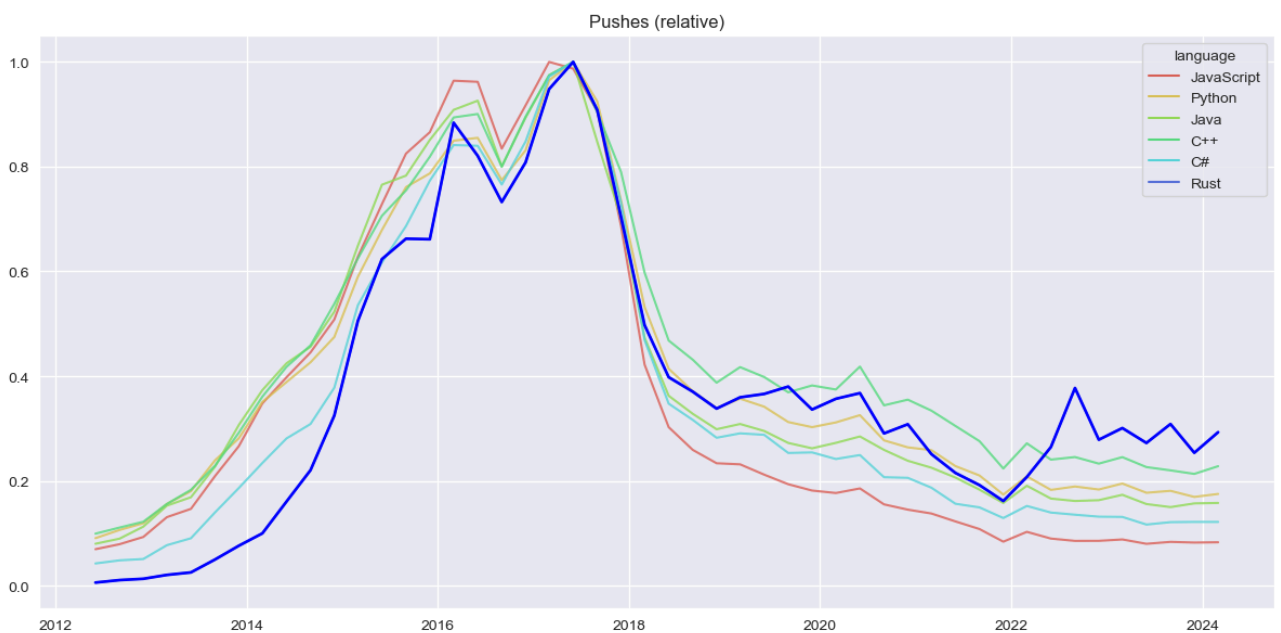


Рисунок 3.9 – Графік відносної кількості pushes у публічних репозиторіях GitHub до пікової кількості для кожної з найбільш популярних мов програмування за даний період

На основі наданих графіків, які відображають різні аспекти активності на GitHub для основних мов програмування з 2012 по 2024 рік, можна зробити наступний загальний аналіз:

1. Загальна тенденція росту (2012-2017):

- Усі мови показують значне зростання активності з 2012 до приблизно 2017 року.
- Це відображає загальне зростання використання GitHub та збільшення кількості проєктів з відкритим кодом.

2. Пік активності (2017-2018):

- Більшість мов досягають піку активності приблизно в 2017-2018 роках.
- Це може вказувати на період максимального розвитку та впровадження цих мов.

3. Спад та стабілізація (2018-2024):

- Після 2018 року спостерігається загальний спад активності для більшості мов.
- Цей спад може бути пов'язаний з насиченням ринку, зміною трендів у розробці або появою нових технологій.

4. Особливості Rust:

- Rust демонструє унікальну траєкторію росту порівняно з іншими мовами.
- Починаючи з найнижчих показників у 2012-2014 роках, Rust показує стрімке зростання.
- До 2024 року Rust досягає або перевищує показники інших мов за більшістю метрик.
- Особливо помітне лідерство Rust у кількості pull requests та stars в останні роки.

5. Стабільність Python:

- Python демонструє стабільно високі показники протягом усього періоду.
- Незважаючи на загальний спад після 2018 року, Python зберігає сильні позиції.

6. Зниження активності JavaScript:

- JavaScript показує найбільш помітний спад активності після 2018

року.

- Це може бути пов'язано з насиченням ринку веб-розробки або зміною трендів.

7. Відносна стабільність C++ та Java:

- C++ та Java демонструють більш помірні коливання порівняно з іншими мовами.
- Вони зберігають середні позиції протягом усього періоду, що може вказувати на їх стабільне використання в усталених галузях.

8. Динаміка C#:

- C# показує помірне зростання до 2018 року, після чого спостерігається поступовий спад.
- Проте, спад C# менш виражений порівняно з JavaScript.

9. Зміна лідерства:

- На початку періоду лідерами за більшістю показників були JavaScript та Python.
- До кінця періоду Rust виходить на передові позиції, особливо за показниками pull requests та stars.

10. Різниця між метриками:

- Графік "Issues" показує більш плавні зміни порівняно з іншими метриками.
- "Pull Requests" та "Pushes" демонструють більш різкі коливання, особливо для Rust.
- "Stars" показують найбільшу волатильність, що може відображати зміни в популярності проектів.

ВИСНОВОК

Дослідження динаміки розвитку та використання мови програмування Rust за період з 2012 по 2024 рік демонструє значний і стабільний ріст її популярності та адаптації в спільноті розробників. Ключові спостереження та висновки включають:

1. Стрімке зростання: Rust продемонстрував найбільш вражаюче зростання серед усіх проаналізованих мов програмування. Починаючи з найнижчих показників у 2012-2014 роках, Rust досяг лідируючих позицій за більшістю метрик до 2024 року.
2. Активність спільноти: Аналіз даних GitHub показує, що Rust має найвищі показники відносної кількості pull requests та stars в останні роки, що свідчить про активну та зацікавлену спільноту розробників.
3. Стійкість тренду: На відміну від інших мов, які показали спад після піку популярності у 2017-2018 роках, Rust продовжив зростання, демонструючи стійкий інтерес та розвиток.
4. Розширення екосистеми: Зростання інтересу до ключових бібліотек Rust, таких як Tokio для асинхронного програмування та Serde для серіалізації даних, вказує на розширення сфер застосування мови та зрілість її екосистеми.
5. Конкурентоспроможність: Rust успішно конкурує з усталеними мовами програмування, такими як Python, JavaScript, Java та C++, особливо в областях, де критично важливі продуктивність та безпека.
6. Потенціал для майбутнього розвитку: Тенденції пошукових запитів та активність на GitHub вказують на зростаючий інтерес до вивчення та використання Rust, що передбачає подальше зміцнення його позицій у майбутньому.
7. Адаптація в індустрії: Використання Rust такими технологічними гігантами, як Mozilla, Microsoft, Google та Amazon, підтверджує його зростаюче значення в промисловій розробці програмного забезпечення.
8. Унікальні переваги: Зростання популярності Rust можна пов'язати з його

унікальними характеристиками, такими як безпека пам'яті без втрати продуктивності, що робить його привабливим для розробки системного програмного забезпечення та високонавантажених застосунків.

Загалом, дослідження переконливо демонструє, що Rust є мовою програмування зі стрімким розвитком та значним потенціалом. Його зростаюча популярність, активна спільнота розробників та розширення сфер застосування вказують на те, що Rust стає все більш важливим гравцем у світі розробки програмного забезпечення. Ці тенденції дозволяють прогнозувати, що Rust продовжить зміцнювати свої позиції та може стати однією з провідних мов програмування в найближчому майбутньому, особливо в галузях, де критично важливі ефективність, безпека та надійність коду.

ПОСИЛАННЯ

1. Офіційний сайт Rust [Електронний ресурс] - <https://www.rust-lang.org/>
2. Google Trends walkthrough [Електронний ресурс] - <https://www.youtube.com/watch?v=GXpGoUw4YUw&list=PLKoqnv2vTMUO8MUpCbVIiBaDZYrpNF8f-&index=4>
3. Офіційни сайт Tokio [Електронний ресурс] - <https://tokio.rs/>
4. Репозиторій бібліотеки Serde [Електронний ресурс] - <https://github.com/serde-rs/serde>
5. Офіційний сайт TIOBE [Електронний ресурс] - <https://www.tiobe.com/tiobe-index/>
6. Офіційний сайт PYPL [Електронний ресурс] - <https://pypl.github.io/PYPL.html>
7. Сторінка midnight.github.io [Електронний ресурс] - <https://madnight.github.io/githut/#/>
8. Офіційний сайт Github [Електронний ресурс] - <https://github.com/>

ДОДАТОК А

Тексти програмного коду
(Найменування програми (документа))

Жорсткий диск
(Вид носія даних)

(Обсяг програми (документа), арк.)

Студента групи ІІІ-ІІ 4 курсу

Сідак К. І


```
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import warnings
```

```
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
def load_data(path: str, languages: list) -> pd.DataFrame:
```

```
    df = pd.read_csv(path)
```

```
    df = df.rename(columns={'name': 'language'})
```

```
    df = df[df['language'].isin(languages)]
```

```
    df['date'] = (df['quarter'] * 3).astype(str).str.zfill(2) + '/' + df['year'].astype(str)
```

```
    df['date'] = pd.to_datetime(df['date'], format='%m/%Y')
```

```
    return df
```

```
def show_trend(df: pd.DataFrame, title: str, target_language: str = 'Rust',
relative=True) -> None:
```

```
    y = 'count'
```

```
    if relative:
```

```
        df['relative_count'] = df['count'] /
```

```
df.groupby('language')['count'].transform('max')
```

```
    y = 'relative_count'
```

```
    sns.set_palette("hls", 8)
```

```
    sns.set_style('darkgrid')
```

```
    plt.figure(figsize=(12, 6))
```

```
    ax = sns.lineplot(data=df, x='date', y=y, hue='language', legend='full',
alpha=0.8)
```

```
    target_df = df[df['language'] == target_language]
```

```
    sns.lineplot(data=target_df, x='date', y=y, linewidth=1.9, color='blue', ax=ax,
legend=False)
```

```
handles, labels = ax.get_legend_handles_labels()
for i, label in enumerate(labels):
    if label == target_language:
        handles[i].set_color('blue')
plt.xlabel("")
plt.ylabel("")
plt.title(title + ' (relative)')
plt.tight_layout()
plt.show()
```

```
languages = ['Rust', 'C++', 'Python', 'Java', 'C#', 'JavaScript']
directory = 'githut-csv/'
```

```
file_path = directory + 'gh-pull-request.json.csv'
pull_requests = load_data(file_path, languages)
show_trend(pull_requests, 'Pull Requests')
```

```
file_path = directory + 'gh-star-event.json.csv'
stars = load_data(file_path, languages)
show_trend(stars, 'Stars')
```

```
file_path = directory + 'gh-issue-event.json.csv'
issues = load_data(file_path, languages)
show_trend(issues, 'Issues')
```

```
file_path = directory + 'gh-push-event.json.csv'
pushes = load_data(file_path, languages)
show_trend(pushes, 'Pushes')
```