

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЗБОРУ КРИПТОВАЛЮТИ

Автоматизація розробки програм є важливим напрямком при підвищенні продуктивності праці. Вона дозволяє поліпшити якість продукції, оптимізувати процеси управління, а також відсторонити людину від виробництв, небезпечних для здоров'я. Автоматизація, за винятком найпростіших випадків, вимагає комплексного та системного підходу і особливих методів до вирішення різних завдань та проблем. Методи обчислень що застосовуються, іноді копіюють нервові і розумові функції людини. Тому весь цей комплекс засобів автоматизації зазвичай називають системами [1].

Важливим напрямком в області автоматизації програмування, крім мов програмування, бібліотек і пакетів прикладних програм, є створення інструментальних засобів розробки програмного забезпечення (ПЗ). При цьому сучасні засоби автоматизації процесу реалізації ПЗ дозволяють скоротити тривалість такого складного процесу, як ведення програмної документації. Цей процес реалізується з використанням генераторів документації програм. Вони аналізують мовні конструкції і коментарі у вихідному коді програм та їх конструкцій [1, 2].

Основний напрямок розвитку систем автоматизації це створення автоматичних систем, що здатні виконувати задані функції чи інші завдання без участі людини. Роль людини полягає у підготовці вихідних даних, виборі алгоритму (методу рішення) і аналізу отриманих результатів. В подібних системах передбачається поступово нарощувати захист від нестандартних подій (аварій) або способи їх обходу [2].

Для автоматизації процесу збору криптовалют розглянемо середовище розробки, мову програмування та систему вирішення Recaptcha v2, а також новітню її версію Recaptcha v3. Метою роботи є реалізація програмного забезпечення для збору криптовалют. Об'єктом дослідження – процес автоматизації збору криптовалют. Предметом – моделі, методи та технології автоматизації збору криптовалют.

Microsoft Visual Studio є серією продуктів фірми Майкрософт, які включають інтегроване середовище розробки програмного забезпечення та ряд інструментальних засобів. Вони дозволяють розробляти консольні програми і програми з графічним інтерфейсом та підтримкою технології

Windows Forms, а також веб-сайти, веб-застосунки, веб-служби як в стандартному, так і в керованому кодах для всіх платформ, що підтримуються [3].

Visual Studio побудована за архітектурою яка підтримує процеси використання доповнень (Add-Ins) чи плагінів від сторонніх розробників, що дозволяє розширювати можливості середовища розробки.

Вона включає редактор вихідного коду з підтримкою технології IntelliSense і можливістю використання найпростішого рефакторінга коду. Вбудований відладчик може працює як налаштувальник рівня вихідного коду або як налаштувальник машинного рівня. Решта інструментів, що вбудовуються, включають в себе редактор форм для спрощення реалізації графічного інтерфейсу додатку, веб-редактор, дизайнер класів і дизайнер схеми бази даних [3].

Visual Studio дозволяє створювати і підключати сторонні додатки з метою розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій та вихідного коду (Subversion і Visual SourceSafe), а також додавання нових наборів інструментів та редагування і візуального проектування коду на предметно-орієнтованих мовах програмування чи інструментів для інших подібних завдань циклу розробки програмного забезпечення (клієнт Team Explorer при роботі з Team Foundation Server) [4].

C# є об'єктно-орієнтованою мовою програмування з безпечною системою типізації для платформи .NET. Синтаксис C# близький до C++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML [5].

Переїнявши багато від мов C++, Delphi, Модула і Smalltalk - C# виключає моделі, які є проблемними при розробці програмних систем, наприклад множинне спадкування класів. У C# існує єдина система типів які використовують загальний набір операцій і їх значення можна зберігати, передавати і обробляти схожим чином [6].

У третій версії Recaptcha вдосконалений поведінковий аналіз, то є та сама система advanced risk analysis, просунутого аналізу ризику. Тепер система працює «в тлі» і непомітно для користувачів.

Досить завантажити разом зі сторінкою бібліотеку Recaptcha і запустити greaptcha.execute в певний момент або відразу в момент завантаження сторінки [3].

З точки зору веб-майстра, головною відмінністю третьої версії є те, що за запитом через API сервер Recaptcha видає не бінарне значення, а оцінку в діапазоні від 0.0 (ймовірний бот) до 1.0 (ймовірна людина) для даного конкретного запиту. Відповідь надсилається у форматі JSON, приклад

наведено у лістингу 1.

Лістинг 1 – Приклад відповіді Recaptcha v3

```
{ "success": true|false, // whether this request was a valid reCAPTCHA token
  for your site
  "score": number // the score for this request (0.0 - 1.0)
  "action": string // the action name for this request (important to verify)
  "challenge_ts": timestamp, // timestamp of the challenge load (ISO format
  yyyy-MM-dd'T'HH:mm:ssZZ)
  "hostname": string, // the hostname of the site where the reCAPTCHA was
  solved
  "error-codes": [...] // optional }
```

Для відправки рекапчі на розпізнавання в CapMonster2, необхідно сформулювати запит, який включає url і sitekey сторінки, а також action. У прикладі в якості параметра action - 'homepage'. Після отримання токена необхідно передати його в функцію верифікації. Так як верифікація може статися в будь-який момент часу, потрібно перехопити запит на отримання токена і у відповіді підмінити його на отриманий токен з CapMonster.

Використання обраних методів реалізації програмного забезпечення включає мову програмування C# та середовище розробки Microsoft Visual Studio 2019, вона є крос-платформеною мовою з дуже зручним середовищем розробки та синтаксисом. Visual Studio дозволяє писати код своєю обраною мовою чи будь-якою з інших найбільш структурованих мов, використовуючи один і той же інтерфейс. Також швидке та якісне вирішення Recaptcha v3 є необхідним, тому обрано CapMonster та створено сніппет для успішної підмани такенну при виконанні верифікації.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Вичугова А.А. Автоматизация процесса разработки программного обеспечения: методы и средства. / А.А. Вичугова // Прикладная информатика. – 2016. – Т.11., 3(63) – С.63-75.
2. Томашевський В.М. Моделювання систем. / В.М. Томашевський – Київ: BHV, 2005. – 349 с.
3. Тапскотт Д., Технология блокчейн / Д. Тапскотт, А. Тапскотт – 2016. – 448 с.
4. Fowler M. Patterns of Enterprise Application Architecture. / М. Fowler – Addison-Wesley, – 2012. – 544 p.
5. Характеристика інструментального ПЗ. Режим доступу: <https://lektsii.org/14-72572.html>.
6. Майлз Р. C# Programming Yellow Book, Cheese Edition. / Р. Майлз – Independently published, 2016. – 216 с.