

**Киричек Г.Г.**

Національний університет «Запорізька політехніка»

**Іванов М.А.**

Національний університет «Запорізька політехніка»

## СИСТЕМА АВТОМАТИЧНОГО ЗБОРУ ТА ВИВЕДЕННЯ КРИПТОВАЛЮТИ

У даній роботі запропоновано модель та алгоритм роботи системи автоматичного рішення капч для збору та виведення криптовалюти. Реалізована система автоматичного збору криптомонет з використанням методу емуляції поведінки людини. Для досягнення поставленої мети використовуються метод збору «крани» з рішенням різних капч. Для визначення способу майнінгу використовуються протоколи: *proof-of-work (PoW)* – «доказ роботи»; *proof-of-stake (PoS)* – «доказ частки»; *proof-of-activity (PoA)* – «доказ активності». У якості програмних методів в роботі обрано мову програмування C#, середовище розробки Microsoft Visual Studio 2017 та програмний комплекс CapMonster. Для розгадування ReCaptcha v2, через HTTP-запити до CapMonster 2 підключаємо додатковий модуль «ReCaptcha2 SiteKey Addon». Метою роботи є реалізація системи для збору криптовалюти. Об'єкт дослідження – процес автоматизації збору криптовалюти. Предмет дослідження – моделі, методи та програмні засоби для розв'язання завдань автоматизації збору криптовалюти. У роботі обрано метод заробітку криптовалюти – кран. Завдання, які треба розв'язати під час роботи, є такими: дослідити та розглянути наявні крани з завданнями на рішення капчі; проаналізувати та вибрати кран, який виплачує і має реферальну систему; змодельовати архітектуру та функціональні складові частини системи автоматичного збору криптовалюти; реалізувати функціональні складові частини та модуль рішення різних видів капч; протестувати реалізований модуль рішення капч. За реалізації багатопотокової програми гарантуємо, що будь-яка частина даних захищена від можливості зміни їхніх значень безліччю потоків. Методи дослідження базуються на моделюванні системи та схем взаємодії її модулів, вимірах основних характеристик системи та характеристик рішень капчі.

**Ключові слова:** recaptcha, криптовалюта, майнінг, біткоїн, кран.

**Постановка проблеми.** Важливим напрямом в області автоматизації програмування, крім мов програмування, бібліотек і пакетів прикладних програм, є створення інструментальних засобів розробки програмного забезпечення. Криптовалюта – звичайні гроші, універсальний товар, який є мірою вартості при купівлі та продажу інших товарів. Вона має декілька суттєвих переваг: їй не страшна інфляція; вона підтримує децентралізованість та анонімність; їй притаманні доступність та безпека, а також глобальність та швидкість [1]. Транзакції в наявних відкритих криптовалютних системах є незворотними. Криптовалюта складається з декількох записів: адреси відправника; суми грошового переказу та адреси одержувача криптовалюти. Водночас учасник криптовалютної мережі (нода), на якому працює її програмне забезпечення, перевіряє усі блоки та транзакції та має функціонал гаманця.

**Аналіз останніх досліджень і публікацій.** Найпоширенішим способом заробітку криптовалюти вважається майнінг, кінцева мета якого – підбір

цифрового підпису, що закриває блок [2]. Для визначення способу майнінгу використовуються протоколи: *proof-of-work (PoW)* – «доказ роботи»; *proof-of-stake (PoS)* – «доказ частки»; *proof-of-activity (PoA)* – «доказ активності». Майнінг за винагороду не лише забезпечує функціонування системи, але й допомагає досягти мережевого ефекту – люди приєднуються до мережі, щоб одержати вигоду [3]. Вибір місця розташування дата центру для майнінгу залежить від трьох основних чинників: доступ до дешевої електроенергії, здатність швидко надсилати та одержувати через Інтернет великі масиви даних, та клімат, який має запобігати перегріву апаратури. Три чверті пулів зосереджені у двох країнах: 58% пулів загального хешрейту мережі Біткоїн, розташовуються у Китаї та 16% у США. Іншими популярними регіонами є Ісландія, Грузія, та Східна Європа. Хешрейт мережі Біткоїну, у діапазоні вересень 2017 – липень 2019, наведено на рис. 1 [4].

Крім майнінгу є багато інших методів заробітку криптовалюти: крани – послуги з роздачі Bitcoin та

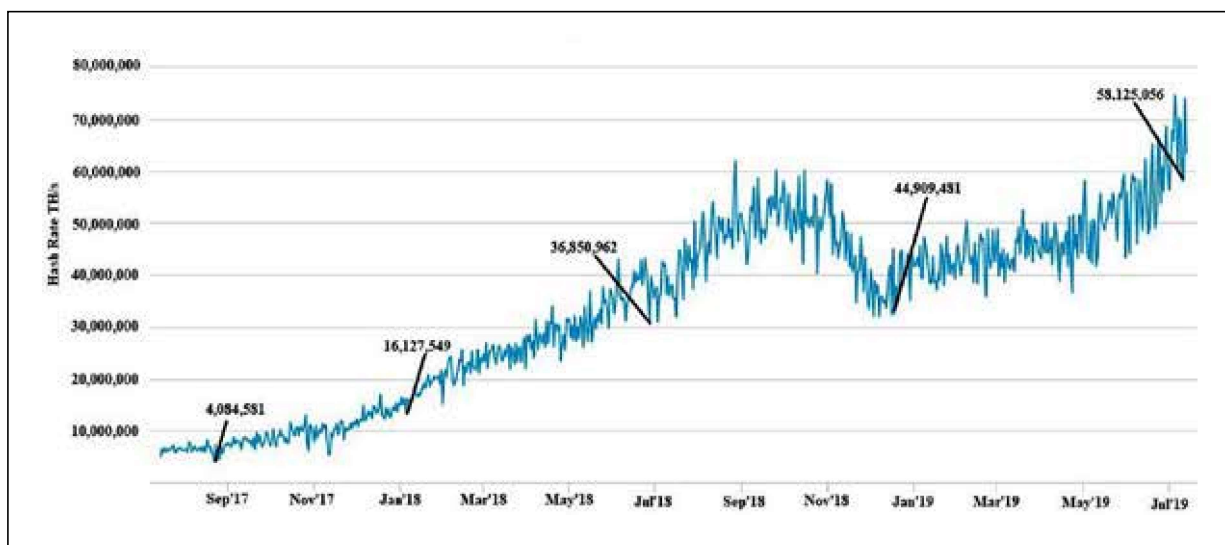


Рис. 1. Хешрейт біткоїну



Рис. 2. Відсотки розпізнавання ReCaptcha v2 та інших видів капч

інших монет за виконання дрібних завдань: розгадування капчі, веб-серфінг тощо; баунті – процес реклами нових ICO-проектів через постинг, репостинг, переклади; постинг/копірайтинг – учасники сервісів Steemit і Golos отримують винагороду у вигляді внутрішньої криптовалюти. Алгоритм, за яким здійснюється безкоштовна роздача криптовалюти, виглядає таким чином: власник ресурсу розміщує банерну і відеорекламу та починає просування свого сайту; відвідувачі сайту, залучені у виплаті криптовалюти, виконують задані дії; після виконання завдання або через певний інтервал часу користувач може вивести зароблені кошти [5].

У якості програмних методів в роботі обрано мову програмування C#, середовище розробки Microsoft Visual Studio 2017 та програмний комплекс CapMonster. C# об'єктно-орієнтована мова

програмування з C-подібним синтаксисом, близьким до C++ і Java. Питання управління версіями істотно вплинули на такі аспекти розробки C#, як роздільні модифікатори virtual і override, правила вирішення перевантаження методів і підтримка явного оголошення членів інтерфейсу [6].

Середовище розробки Microsoft Visual Studio є зручною крос-платформною мовою з висококласним середовищем розробки та синтаксисом [6; 7]. CapMonster – програмний комплекс, призначений для розпізнавання капч, має три версії: professional – максимальна кількість одночасних потоків розгадування капчі дорівнює двадцяти та більше; standart – обмеження кількості потоків розгадування капчі до 5; lite – версія з однопоточним розгадуванням різних видів капч [8]. Відсотки розпізнавання ReCaptcha v2 та інших видів капч зображено на рис. 2.

Для розгадування ReCaptcha v2 через HTTP-запити треба приєднати до CapMonster 2 додатковий модуль «ReCaptcha2 SiteKey Addon» [9].

Далі проведемо вибір рішень і технологій для реалізації системи автоматичного збору та виведення криптовалюти.

**Постановка завдання.** Метою роботи є реалізація системи для збору криптовалюти. Об'єкт дослідження – процес автоматизації збору криптовалюти. Предмет дослідження – моделі, методи та програмні засоби для розв'язання завдань автоматизації збору криптовалюти.

У роботі обрано метод заробітку криптовалюти – кран. Завдання, які треба вирішити під час роботи, є такими: дослідити та розглянути наявні крани з завданнями на рішення капчі; проаналізувати та вибрати кран, який виплачує і має реферальну систему; змодельовати архітектуру та функціональні складові частини системи автоматичного збору криптовалюти; реалізувати функціональні складові частини та модуль рішення різних видів капч; протестувати реалізований модуль рішення капч [2; 3].

За реалізації багатопоточної програми гарантуємо, що будь-яка частина даних захищена від можливості зміни їхніх значень безліччю потоків. Зважаючи, що всі потоки в AppDomain мають паралельний доступ до даних програми, очевидно, що доступ до таких загальних даних необхідно синхронізувати. Це гарантує, що в один момент часу доступ до певного блоку коду отримає лише один потік. Так ми гарантуємо цілісність даних, а також їхню актуальність в будь-який момент часу. З варіантів синхронізації, зазвичай використовують 4 види: блокування виклику коду; конструкції, що обмежують доступ до частин коду; сигналізуючі конструкції та неблокуюче блокування [10].

Методи дослідження базуються на моделюванні системи та схем взаємодії її модулів, вимірах основних характеристик системи та характеристик рішень капчі [11]. Для зберігання криптовалюти необхідно мати електронний гаманець. У гаманці зберігаються не самі монети, а приватні ключі, які дозволяють отримати доступ до рахунків у блокчейні. Децентралізацію майнінгу криптовалют можна посилити з допомогою майнінгових завдань, розв'язання яких не можливо прискорити з допомогою спеціалізованих інтегральних схем (ASICs).

**Виклад основного матеріалу.** Центральним елементом реалізації програмного забезпечення є моделювання. Моделі дозволяють наочно про-

демонструвати потрібну структуру і поведінку системи. Моделі допомагають досягти кращого структурування модулів системи, що часто призводить до їхнього спрощення та можливості повторного використання [12].

Для роботи desktop-додатку у діаграмах варіантів використання описані такі актори: користувач – особа, що завантажила додаток з метою збору криптовалюти; програміст – людина, яка займається розробкою програмного забезпечення (рис. 3) [13].

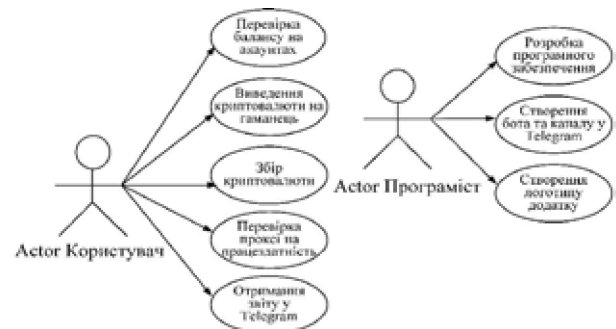


Рис. 3. Діаграми варіантів

Діаграма послідовності показує життєвий цикл об'єкту і взаємодію акторів. Опис роботи користувача у додатку зі збору криптовалюти, зображує послідовність дій. Основний виконавець: користувач.

Основний потік: користувач запускає додаток; користувач вводить дані у поля для заповнення інформації про акаунт; користувач зазначає необхідні файли для подальшої роботи та спостерігає результати збору криптовалюти.

Діаграми послідовності прецедентів «збір криптовалюти» та «виведення криптовалюти на електронний гаманець» зображено на рис. 4 [13; 14].

Подібна діаграма отримання звіту про збір криптовалюти у Telegram наведена на рис. 5.

Для роботи з системою реєструємо облікові записи на freedoge.co.in. Після чого вставимо список акаунтів у відповідне поле за форматом, який вказаний в програмі. Також потрібно вставити список проксі для кожного акаунта і вибрати тип (http, socks4, socks5). Для обмеження кількості потоків програми задаємо параметр в поле «Threads». Коли акаунт розгадав рекапчу та зібрав криптовалюту йому назначаться таймер на сайті, (програма чекає відповідний час, який встановлений для акаунта), число в полі «Wait timer». Якщо ставимо 500, то програма очікує акаунт до 500 секунд. Під час розробки програми потрібно увімкнути бібліотеки (лістинг 1).

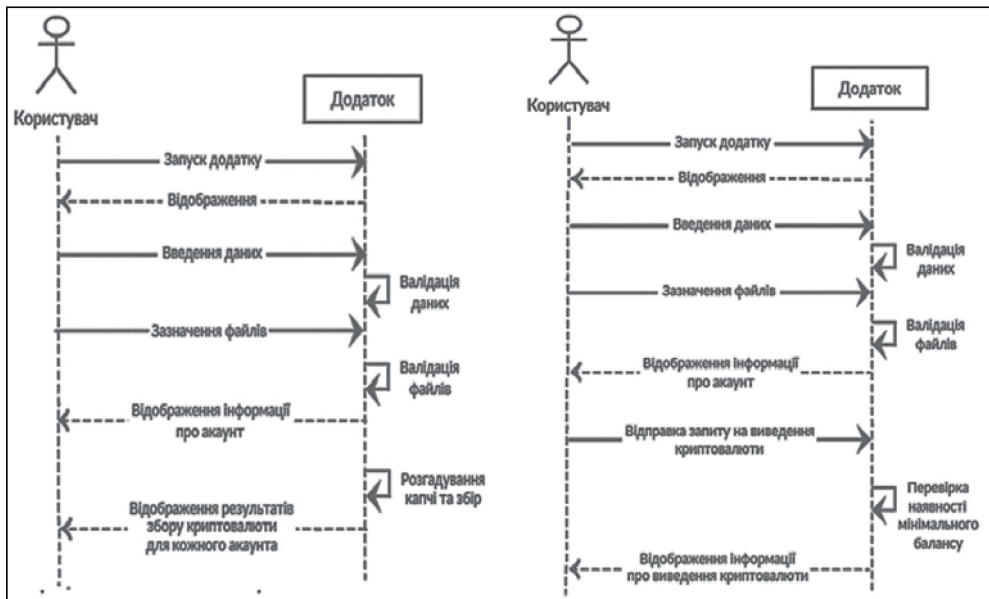


Рис. 4. Діаграми послідовності прецедентів «збір криптовалюти» та «виведення криптовалюти на електронний гаманець»

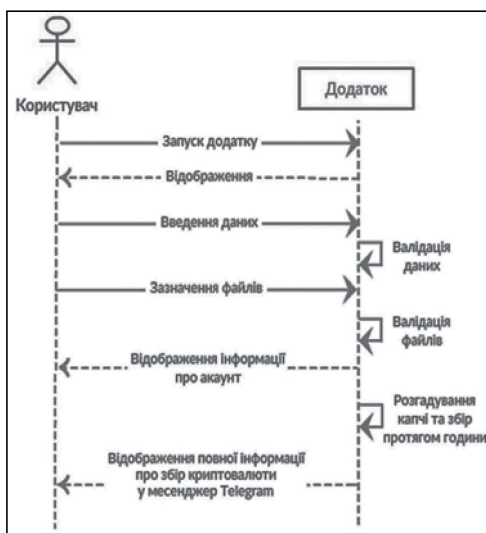


Рис. 5. Діаграма послідовності прецеденту «отримання звіту у Telegram»

Лістинг 1 – Підключення бібліотек:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Threading;
using System.Windows.Forms;
using xNet;
using Telegram.Bot;

```

Для запуску збору криптовалюти у «Start» реалізуємо код, що наведено у лістингу 2.

Лістинг 2 – Функція старту програми:

```

public void button1_Click(object sender,
EventArgs e)
{
    richTextBox1.Text = richTextBox1.Text.
TrimEnd();
    kolvo_strok = richTextBox1.Lines.Count();
    label3.Text = "[" + kolvo_strok.ToString() + "]";
    int threadCount = Convert.ToInt32(textBox2.
Text);
    //Create and start threads
    for (int i = 0; i < threadCount; i++)
        new Thread(Process) { IsBackground = true
    }.Start();
}

```

Після натискання кнопки «Start» перевіряються усі поля та файли, на заповнення (лістинг 3).

Лістинг 3 – Перевірка на заповнення усіх полів та файлів:

```

if ((openFileDialog1.FileName == String.Empty)
|| (openFileDialog2.FileName == String.Empty) ||
(openFileDialog3.FileName == String.Empty))
{
    MessageBox.Show("Choose the path to the
files!", "Wrong path(s)!");
}
else
{
    //continue working...
}

```

Якщо всі поля заповнені і файли вказані вірно: виконуються подальші команди для підготовки акаунтів та GET-запит для отримання додаткових даних. Для збору криптовалюти виконуємо POST-запит з параметрами акаунту: спочатку розпізнаємо рекапчу з допомогою

sitekey (відповідь GET-запиту); відправляємо GET-запит на розгадування рекапчі з параметрами sitekey та адресою сайту; отримуємо ідентифікатор капчі та відповідь від CapMonster і відправимо GET-запит на отримання результату: `http://127.0.0.1/in.php?key=&method=userrecaptcha&googlekey=sitekey&pageurl=https://freedoge.co.in; http://127.0.0.1/res.php?action=get&id=CapchaID`. Отримання результату капчі відбувається з допомогою коду, що наведено у лістингу 4.

Лістинг 4 – Отримання результату Recaptcha v2:

```
static SemaphoreSlim semaphoreSlim = new
SemaphoreSlim(1, 1);
string recaptcha = string.Empty;
while (recaptcha == String.Empty)
{
    await semaphoreSlim.WaitAsync();
    try
    {
        List<string>quotelist=File.
ReadAllLines(captcha_folder).ToList();
        recaptcha = quotelist[0];
        quotelist.RemoveAt(0);
        File.WriteAllLines(captcha_folder,    quotelist.
ToArray());
    }
    catch { await Task.Delay(2000); }
    finally
    { semaphoreSlim.Release(); }
}
```

Після отримання результату розгаданої капчі програма посилає POST-запит на сайт з параметрами акаунту та вирішеною капчею.

Щоб зручно контролювати процес збору в системі створено бота для месенджера Telegram. Він кожну годину повідомляє о кількості зборів за сесію/за годину та кількості криптовалюти за сесію/ за годину (лістинг 5).

Лістинг 5 – Посилання звіту з інформацією про збір криптовалюти кожну годину у месенджер Telegram:

```
lock (time)
{if (Convert.ToInt32(end_time) > Convert.
ToInt32(start_time))
{
    start_time = DateTime.Now.ToString("hh");
    string strChat = "-25658969";
    var test = new TelegramBotClient("675252248:A
AEKCz_ZO-WDfysaulG0RW6yBcCgm3JrPy");
    var GetMe = test.GetMeAsync();
    var SendMessage = test.
SendMessageAsync(strChat, "Freedoge Bot ★
Зборів за сесію:" + kolvo_sbor + " раз. Зборів за
годину: " + kolvo_sbor_hour + " раз"); }
else if (start_time == "12" && end_time == "01")
{
    start_time = DateTime.Now.ToString("hh");
```

```
string strChat = "-25658969";
var test = new TelegramBotClient("675252248:A
AEKCz_ZO-WDfysaulG0RW6yBcCgm3JrPy");
var GetMe = test.GetMeAsync();
var SendMessage = test.
SendMessageAsync(strChat, "Freedoge Bot ★
Зборів за сесію:" + kolvo_sbor + " раз. Зборів за
годину: " + kolvo_sbor_hour + " раз");
}}
```

Після завершення реалізації системи перевіряємо її працездатність. Для цього реєструємо акаунт на сайті крапа криптовалюти. Далі заповнюємо поле «Accounts». У вкладці «Cookies» вказуємо директорію файлу з куками для акаунту (наприклад, «D:\Cookies») та вибираємо тип проксі(в системі обрано «Http»). Останнім етапом налаштування є зазначення трьох файлів: файлу з вирішеними капчами; файлу з резервним списком проксі; файлу для акаунтів, які потребують верифікацію через пошту.

Перед початком роботи системи (після налаштування) натискаємо кнопку «Start». У вкладці «Info» можна побачити шість стовпців з певною інформацією: перший «Accs» відбиває пошту, на яку зареєстровано акаунт; другий «Info» надає інформацію, яку надсилає сайт як відповідь на запит; третій «Balance» інформує про баланс акаунта; четвертий «Timer» виводить кількість секунд до наступного збору; п'ятий «Date» показує час, коли акаунт зібрав криптовалюту чи зробив іншу дію; шостий «Roll №» відбиває кількість збору за всю сесію роботи системи.

Щоб побачити детальну інформацію про роботу системи, потрібно відкрити вкладку «Log». Повний звіт про збір криптовалюти приходить кожну годину у месенджер Telegram.

Результатом тестування програми є повний робочий функціонал та звіт про успішний збір криптовалюти у месенджері Telegram.

**Висновки.** Отже, у роботі наведено аналіз криптовалюти, її переваги та недоліки. Проаналізовано та обрано метод збору криптовалюти. Запропоновано модель та алгоритм роботи системи автоматичного рішення капч для збору та виведення криптовалюти на гаманець. Реалізоване програмне забезпечення автоматичного збору криптомонет з емуляцією поведінки людини, проведено аналіз результатів реалізації програмного забезпечення автоматичного збору криптовалюти, розглянуто метод збору криптовалюти у багатопоточному режимі з використанням проксі та автоматичним рішенням ReCaptcha v2.

**Список літератури:**

1. Kirichek G., Tymoshenko V., Rudkovskiy O., Hrushko S. Decentralized System for Run Services. *CMIS*. 2019. P. 860–872.
2. Поппер Н. Цифровое золото : невероятная история Биткойна. Москва : Вильямс, 2016. 368 с.
3. Тапскотт Д., Тапскотт А. Технология блокчейн. Москва : Эксмо, 2016. 448 с.
4. Hash Rate. The estimated number of tera hashes per second. URL: <https://www.blockchain.com/charts/hash-rate?timespan=2years>(дата звернення: 10.07.2019).
5. Vigna P., Casey M.J. The Age of Cryptocurrency : How Bitcoin and the Blockchain Are Challenging the Global Economic Order. London : St. Martin's Press, 2015. 368 p.
6. Майлз Р. C# Programming Yellow Book, Cheese Edition. Independently published, 2016. 216 с.
7. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. Санкт Петербург : Питер, 2013. 896 с.
8. CapMonster. URL: <http://zennobot-shop.ru/capmonster/> (дата звернення: 10.07.2019).
9. Распознавание ReCaptcha 2 через sitekey. URL: <https://zennolab.com/wiki/ru:addons:capmonster:sitekey> (дата звернення: 10.07.2019).
10. Основы многопоточности в .NET Framework. URL: <https://habr.com/ru/company/nixsolutions/blog/260745> (дата звернення: 10.07.2019).
11. Томашевський В.М. Моделювання систем. Київ : BHV, 2005. 349 с.
12. Vaghela V. B., Jadav B. M., Scholar M. E. Analysis of various sentiment classification techniques. *International journal of Computer applications*, 2016. T.140. №. 3. P. 0975–8887.
13. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. Москва : ДМК Пресс, 2006. 496 с.
14. Ларман К. Применение UML 2.0 и шаблонов проектирования. Москва : Вильямс, 2013. 736 с.

**Kirichek G.G., Ivanov M.A. THE SYSTEM OF AUTOMATIC COLLECTION AND CRYPTOCURRENCY WITHDRAWAL**

*In the paper proposes the model and algorithm of work system the automatic solution captch for collection and cryptocurrency withdrawal. The system of automatic collection of cryptomonets with the use of the method of human behavior emulation was implement. To achieve the goal, the method of collecting “cranes” with the decision of different captch is using. Protocols are used to determine the way of mining: proof-of-work (PoW) – “proof of operation”; proof-of-stake (PoS) – “proof of particle”; proof-of-activity (PoA) – “evidence of activity”. As a software method, C # programming language, Microsoft Visual Studio 2017 development environment, and CapMonster software package were select. For unravel ReCaptcha v2, we add an additional module “ReCaptcha2 SiteKey Addon” through HTTP requests to CapMonster 2. The purpose of the work is to implement a system for cryptocurrency collection. The object of the study is the process of automating of the cryptocurrency collection. Subject of research is models, methods and software for tasks solving automation of collections and cryptocurrency withdrawal. In the work the method of earning cryptocurrency was choose – crane. The tasks to be solved in the course of work are as follows: to study and consider existing cranes with tasks for the decision of captcha; analyze and select the crane that pays and has a referral system; to simulate the architecture and functional components the system of automatic cryptography collection; implement functional components and solution module of various types of captcha; test the implemented of captch solution module. When implementing a multithreaded program, we guarantee that any part of the data was been protected from the possibility of changing their values in a multitude of threads. The research methods are basing on modeling of the system and schemes of interaction of its modules, measurements the main characteristics of system and the characteristics of captcha solutions.*

**Key words:** recaptcha, cryptocurrency, mining, bitcoin, crane.