

УДК 378.21

А. А. Яровий¹, С. В. Барабан¹, Р. В. Криночкін¹

ІНТЕЛЕКТУАЛЬНИЙ МОДУЛЬ БРАУЗЕРНОЇ СИСТЕМИ УПРАВЛІННЯ ІТ-ПРОЕКТАМИ TRELLO

¹Вінницький національний технічний університет

Анотація. В даній статті запропоновано програмне рішення у вигляді інтелектуального модуля для розширення функціоналу браузерної системи управління ІТ-проектами Trello на основі використання її API. Програмування здійснено мовою python з використанням поширених бібліотек. Для відбору даних з дошок, списків, карток Trello були розроблені функції за допомогою використання методів процедурного програмування. Під час розробки інтелектуального модуля проведено класову декомпозицію програмного коду, розроблено UML-діаграму класів інтелектуального модуля відповідно до існуючої бібліотеки py-trello. Для агрегації даних, що були відібрані з дошок, списків, карток Trello використано сучасну python бібліотеку Pandas. При цьому застосовувалися такі функції Pandas як зчитування, запис до файлу у форматі Microsoft Excel, групування даних, перезавантаження індексу таблиці, арифметичні додавання і множення. Апробація результатів роботи розробленого інтелектуального модуля системи управління Trello підтвердила доцільність розробки при здійсненні щомісячної звітності та обчислення метрик продуктивності роботи працівників.

Ключові слова: Trello, py-trello, python, pandas, API, система управління, функція, ООП, дошка, список, картка, агрегація даних.

Анотация. В данной статье предлагается программное решение в виде интеллектуального модуля для расширения функционала браузерной системы управления ИТ-проектами Trello на основе использования ее API. Программирование проведено на языке python с использованием распространенных библиотек. Для отбора данных из досок, списков, карточек Trello были разработаны функции посредством использования методов процедурного программирования. При разработке интеллектуального модуля проведена классовая декомпозиция программного кода, разработана UML-диаграмма классов интеллектуального модуля в соответствии с существующей библиотекой py-trello. Для агрегации данных, которые были отобраны из досок, списков, карточек Trello использовано современную python библиотеку Pandas. При этом применялись такие функции Pandas как считывания, запись в файл в формате Microsoft Excel, группировка данных, перезагрузка индекса таблицы, арифметические сложения и умножения. Апробация результатов работы разработанного интеллектуального модуля системы управления Trello подтвердила целесообразность разработки при осуществлении ежемесячной отчетности и вычисления метрик производительности работы сотрудников.

Ключевые слова. Trello, py-trello, python, pandas, API, система управления, функция, ООП, доска, список, карточка, агрегация данных.

Abstract. In the article was proposed a programmatic solution in the form of an intelligent module for expanding the functionality of a browser-based management system for IT-projects Trello based on the use of its API. To select data from the boards, lists, cards, functions were developed using procedural programming methods. During the development of the intellectual module, a UML-class diagram of the intellectual module developed according to the existing py-trello library. For aggregation of data, selected from the boards, lists, cards, the modern Pandas library is used. The Pandas features such as reading, writing to a Microsoft Excel file, data grouping, reloading the table index, arithmetic adding and multiplication were used. Approval of the results confirmed the feasibility of the development of the implementation of monthly reporting and calculation of employee productivity metrics.

Key words: Trello, py-trello, python, pandas, API, control system, function, OOP, board, list, card, data aggregation.

DOI: <https://doi.org/10.31649/1999-9941-2018-42-2-49-54>.

Вступ

Система управління проектами є невід'ємною складовою сучасної розробки програмних продуктів і використовується не тільки в ІТ-компаніях різної величини, а й більшістю фрілансерів [1]. Таке розповсюдження стало можливим завдяки наданню проектно-орієнтованого робочого простору для ведення одного чи більше проектів, при цьому доступ є у всіх учасників проекту. Систем управління проектами є надзвичайно багато, з різними можливостями, характеристиками і для будь-якого користувача з будь-яким бюджетом, адже навіть платні системи надають хоча й обмежені, але freeversionвласного продукту. Все одно найбільш перспективним є web-basedсистеми, не дивлячись на те, що компільовані варіанти в деяких випадках більш продуктивні, однак менш гнучкі для кастомізації.

Найбільш поширеною в даний час системою управління проектами є Jiraкомпанії Atlassian, хоча вона розроблялася з самого початку як багтрекер – система відстеження помилок, але поступово доповнилася функціями організації спілкування з користувачами та управління проектами. Таким чином, вона підходить для керування проектами та помилками в коді, але вона платна. Існує безкоштовна версія, проте значно обмежена у функціоналі. Jira підходить для великих ІТ-компаній, а для малих компаній або просто команди програмістів не є оптимальним варіантом. Тому попрацювавши з Jira, ІТ-компанія SinaiR&D вирішила перейти на щось інше, а саме Trello[2]. Trello – безкоштовна браузерна система управління проектами, що використовує парадигму керування проектами, відому як канбан. Проте, щоб повністю замінити Jira, необхідно доробити обгортку навколо Trello, чому і присвячена дана стаття.

Мета

Метою роботи є розширення функціоналу браузерної системи управління ІТ-проектами Trello за рахунок розробки інтелектуального модуля до API Trello.

Задачі

- 1) відбір даних з дошок, списків і карток Trello;
- 2) агрегація отриманих даних;
- 3) виведення даних у вигляді таблиць у файлі формату MicrosoftExcel;
- 4) розробка інтелектуального модуля до API Trello;
- 5) апробація результатів розробки в діяльності IT-команди програмістів SinaiRnD.

Розв'язання задач

Trello є прикладом реалізації простої системи канбан. Канбан було розроблено ученим Таїті Оно, в Toyota, з метою досягнення та підтримки високого рівня виробництва. Канбан став ефективним інструментом в управлінні системою виробництва у цілому, і довів себе, як відмінний спосіб пропагування вдосконалень. Картки є ключовим компонентом канбану і сигналізують про необхідність переміщення задач усередині проекту. Згідно цієї системи у Trello є дошки, списки і картки. Інтелектуальний модуль повинен відбирати назви дошок, назви груп, в які входять дошки, назви карток, ідентифікаційні номери карток, повні імена членів карток, задані користувачем поля в картках [3]. Назви дошок в даному випадку – це назви проектів, групи, в які входять дошки – це назви організацій проектів, назви карток – це є задачі, виконання яких призводить до прогресу проекту, повні імена членів карток – це імена розробників, що виконують поставлені задачі. Найбільш інформативним видом даних є користувацькі поля в картках. Для обліку об'єму зусиль, що знадобились для виконання задачі певного проекту використовується поле – "SP" ("Story Points"), яке являє собою список з чисел Фібоначчі від 1 до 100, а також поле "Payed", що використовується для полегшення фільтрації задач, які вже оброблені системою. Поле "SP" необхідне для нарахування балів розробникам за виконані задачі, на основі чого після агрегації даних має нараховуватися зарплата. "Payed" необхідне для перевірки чи було оплачене конкретне завдання розробнику.

Бібліотека `py-trello`[4], яку ми вибрали за основу для нашої розробки побудована на основі класів. Об'єктами системи Trello є `board`, `list`, `card`, `organization`, `member`, які є екземплярами класів `Board`, `List`, `Card`, `Organization`, `Member` бібліотеки `py-trello`. Для процедурного програмування [5] ми створимо змінні `board`(дошка), `list`(список), `card`(карточка), `organization`(група дошок), `member`(розробник), а також розробимо функції, які будуть використовувати ці змінні та методи класів бібліотеки `py-trello`. Найперша функція програмного модуля має отримувати дошку користувача, а повертати усі картки цієї дошки, що знаходяться у списку виконаних ("Done"). Схему алгоритму роботи даної функції зображено на рис. 1. Функція `list_lists()` є полем класу `Board` бібліотеки `py-trello`, а функція `list_cards()` є полем класу `Card` (рис. 1). Наступна функція перевіряє чи співпадає ідентифікаційний номер користувача картки з ідентифікаційним номером користувача дошки і якщо присутня така рівність, то повертає змінну з даними користувача. Використовує ця функція дошку та ідентифікаційний номер, в якості вхідних даних, а також метод класу `Board.get_members()`. У випадку не співпадіння ідентифікаційних номерів користувача картки і дошки, функція повертає `None`. Дану функцію можна розширити або написати іншу, що повертатиме повне ім'я користувача картки.

Далі потрібно провести роботу з полями користувача в картках. У класу `Card` є метод `get_custom_field_by_name()`, що повертає дані `customfield` картки за іменем. Було створено дві функції, що по реалізації майже ідентичні – `get_sp()` і `get_payed()`. `get_sp()` – приймає картку, використовує функцію `get_custom_field_by_name()` і повертає змінну, в якій записано значення поля "SP" в картці. Якщо в полі "SP" якимось чином опиниться значення `None`, функція поверне цифру 0. Функція `get_payed()` на вході приймає картку, використовує функцію `get_custom_field_by_name()` і повертає `TRUE`, якщо в мітці "Payed" картки стоїть галочка або повертає пустий рядок в іншому випадку.

Для агрегації даних використовуються популярні бібліотеки для роботи з масивами даних `NumPy` та `Pandas` [6]. Перед початком роботи необхідно імпортувати ці бібліотеки до програми. Вони були імпортовані як `np` та `pd` відповідно. В основному тілі програми створено три `DataFrame` – один для даних, відібраних з дошок, списків, карток Trello, другий – для агрегованих даних, а третій `DataFrame` – проміжний, для зберігання даних з `excel`-файлу, що містить вагові коефіцієнти працезатрат за проекти. Кінцевий варіант першого `DataFrame` має вигляд:

Таблиця 1 – `DataFrame`, що утворюється внаслідок виконання програми

	Assignee	Key	Points	Project	Summary
0	Roman	54321abcd	5	demo	Check if PyTrello supports Custom Fields
1	Serhii	145abd	13	demo	Getting known of Python API
2	Serhii	125acd	5	demo	Userstories
3	Serhii	15ad	55	demo	Printtablewith DATA

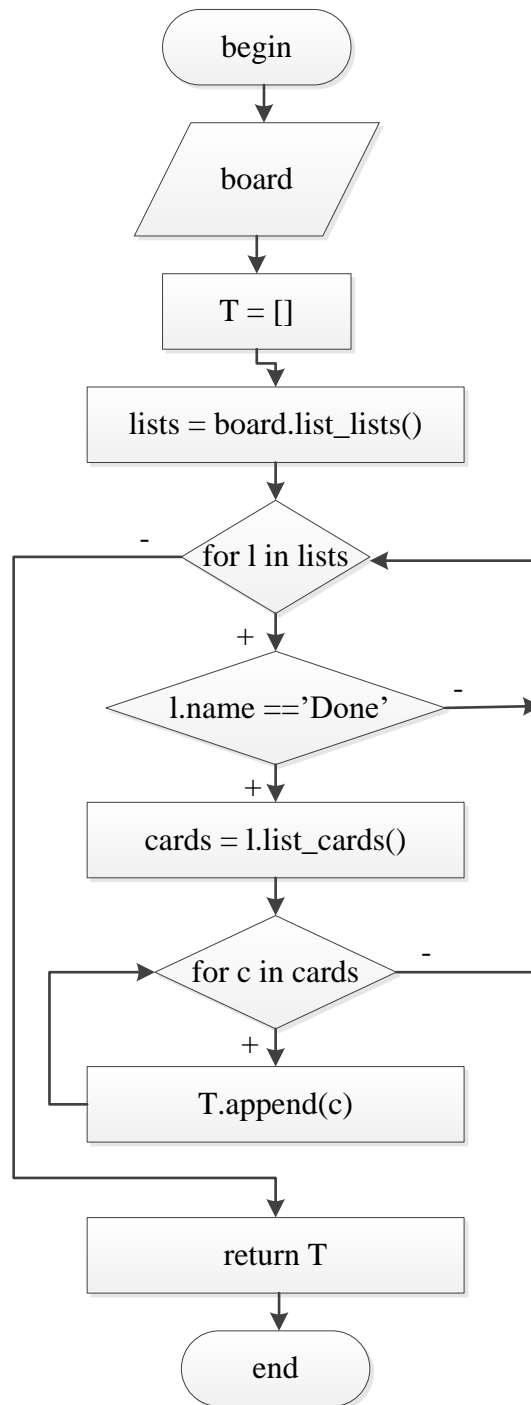


Рисунок 1 – Схема алгоритму роботи функції відбору карток з дошки

Для агрегації даних були використані наступні стандартні функції з бібліотеки Pandas: групування – `groupby()`, знаходження арифметичної суми даних – `sum()`, повернення індексів таблиці до попереднього стану – `reset_index()`, об'єднання декількох `DataFrame` – `merge()`. Зручністю програми є те, що всі функції можна використати в одному рядку, т.зв. “one-liner”:

```
aggregation = array[['Project', 'Assignee', 'Points']].groupby(['Assignee', 'Project']).sum().reset_index()
```

Зчитування даних з excel-файлу в Pandas бібліотеці також досить просте:

```
table = pd.read_excel('ProjectUCP.xlsx', 'Лист1')
```

Кінцевий варіант таблиці з агрегованими даним буде має вигляд:

Таблиця 2 – DataFrame з агрегованими даними

	Assignee	Project	Points	Coeff	Value
0	Serhii	demo	15	5,5	82,5
1	Roman	demo	5	5,5	27,5

Програмний код запис до excel-файлу в декілька листів буде мати вигляд:

```
writer = pd.ExcelWriter('third.xlsx')
array.to_excel(writer, "June")
aggregation.to_excel(writer, "June_agregation")
writer.save()
```

Проектування інтелектуального модуля за методологією ООП

Бібліотека ru-trello, яку ми використовуємо за основу в розробці нашого програмного модуля і яка є обгорткою навколо API Trello, написана на мові програмування Python, є об'єктно-орієнтованою. Тому розробка програмного модуля на основі класів є найбільш логічним вирішенням поставленої задачі.

Першим варіантом є створення класу DataFromTrello(), який би містив в якості методів усі ті функції, що були розроблені вище. Частина коду для розробленого класу DataFromTrello() буде мати вигляд:

```
class DataFromTrello():
def __init__(self):
self.client = TrelloClient(api_key="...", api_secret="...", token="...")
self.boards = self.client.list_boards(board_filter='starred')
self.array = pd.DataFrame(np.object, index=[], columns=[])
```

Далі код класу DataFromTrello() буде містити всі ті ж функції, розроблені для ru-trello за методологією функціонального програмування. Створення екземпляру класу(об'єкту) в тілі програми:

```
a = DataFromTrello()
```

Використання екземпляру класу DataFromTrello() в тілі програми:

```
for b in a.boards:
c = a.cards_from_board(b)
for card in c:
s = a.sp_custom_field(card)
m = a.name_member(b, card)
payed = a.payed_custom_field(card)
```

У цієї програми буде одна особливість. Клас DataFromTrello() не містить ніяких полів. Це означає, що він є статичним класом. Використання статичних класів для вирішення подібних до нашої задачі не є типовим для мови програмування Python[7], а також для методології ООП розробки програмного забезпечення. Тому це не є оптимальним вирішенням задачі розробки програмного модуля системи управління проектами Trello.

Можливим варіантом розробки програмного модуля є використання наслідування класів Board і Card. Тоді постає необхідність створення обгортки для класу TrelloClient.

Більш оптимальним варіантом є використання в тілі програми об'єктів board і card, які будуть екземплярами класів BoardTrello і CardTrello відповідно. На початку розробки програмного модуля було створено UML-діаграму для відображення усіх класів з їх полями і методами для вирішення поставленої у меті задачі. На рис. 2 зображено дану діаграму.

З рис. 2 видно, що клас TrelloClient знаходиться ніби окремо від усіх інших і не пов'язаний з ними, але це клас, що містить поля і методи аутентифікації клієнта системи управління Trello. Напряму він не пов'язаний з усіма класами, але без нього неможлива робота ні одного класу. Класи Board, Card і Member наслідують клас TrelloBase. Таким чином будь-яка зміна класу TrelloBase призведе до необхідності зміни в усіх класів, що його наслідують. Відношення класу BoardTrello до класу Board так як і класу CardTrello до класу Card є композиція, оскільки BoardTrello є частиною цілого Board, а CardTrello є лише частиною цілого Card. При цьому клас BoardTrello залежить від класів Card і Member, а клас CardTrello залежить від класу Member. Відношення залежності зображено на рис. 2 пунктирною лінією. В основному тілі програми утворюються два об'єкти – об'єкт board, що є екземпляром класу BoardTrello та об'єкт card, що є

екземпляром класу CardTrello. Між даними двома об'єктами в тілі програми встановлюється тісний зв'язок. Саме тому відношення між класами BoardTrello і CardTrello є асоціацією. Точніше - відношення між класами BoardTrello і CardTrello є N-арною асоціацією, оскільки одному об'єкту board відповідає множина об'єктів card. Тому на рис. 2 класи BoardTrello і CardTrello з'єднані суцільною лінією, а потужність зв'язку у них 1..*.

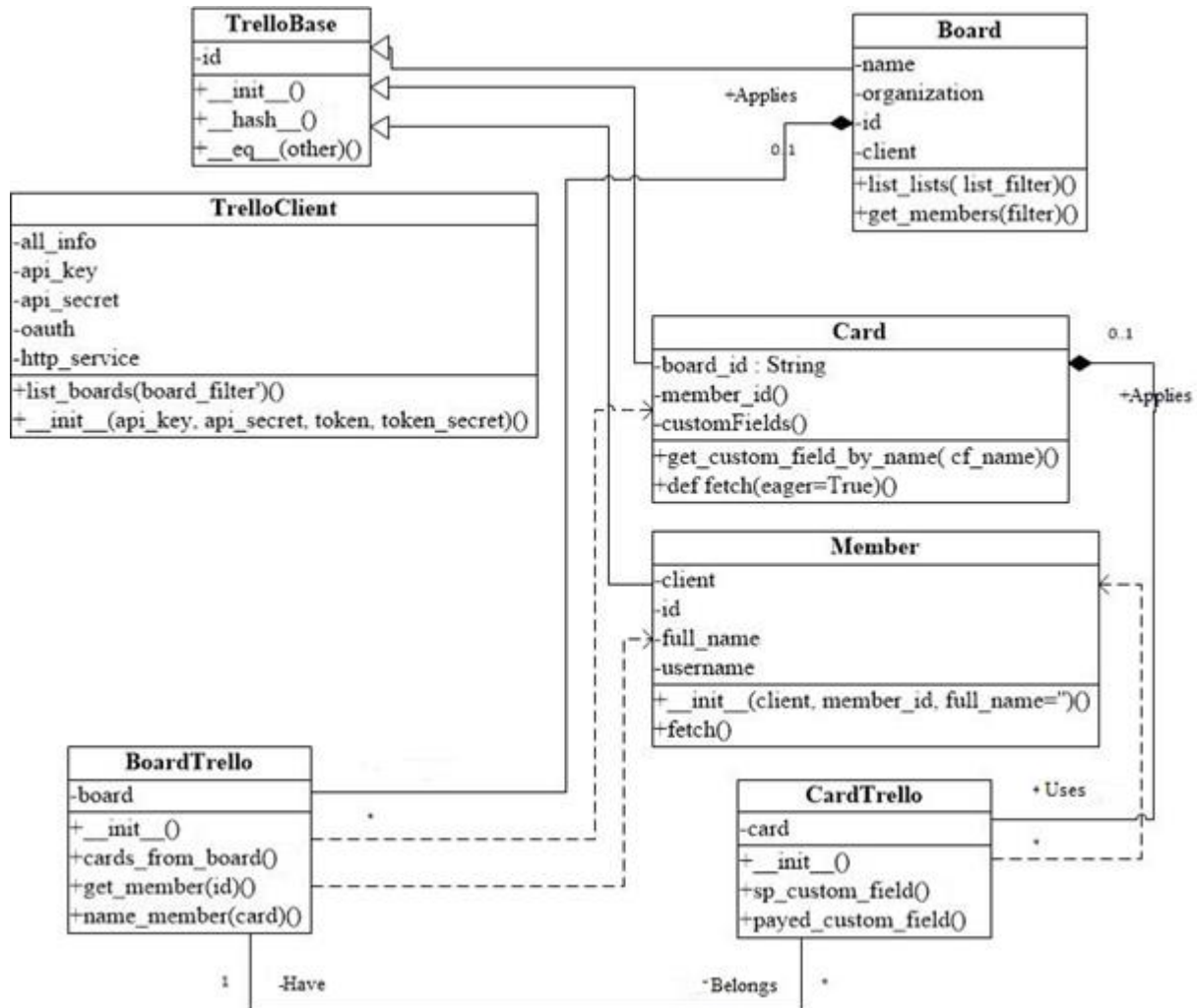


Рисунок 2 – UML-діаграма класів розробленого програмного модуля

Передача board у клас BoardTrello:

```

class BoardTrello():
def __init__(self, board):
    self.board = board
  
```

Обов'язково при передачі board і card необхідно включити перевірку чи ми отримали дійсно board з класу Board і card з класу Card, а якщо ні, то що за тип даних в отриманих board і card. На мові програмування Python це досить легко робиться:

```

assert isinstance(board, Board), "Wrong board type, while expected Board: {}".format(type(board))
  
```

це для board, а для card:

```

assert isinstance(card, Card), "Wrong card type, while expected Card: {}".format(type(card))
  
```

Тоді виклик екземплярів класів BoardTrello і CardTrello можна здійснити безпосередньо в тілі програми в місці, де вони будуть використовуватися:

```

for b in boards:
  
```

```

a = BoardTrello(b)
c = a.cards_from_board()
for card in c:
    d = CardTrello(card)
    s = d.sp_custom_field()
    m = a.name_member(card)
    payed = d.payed_custom_field()

```

Після чого звісно ж йде заповнення DataFrame і запис, отриманих даних до Microsoft Excel, завдяки можливостям Python бібліотеки Pandas.

Висновки

1. Запропоновано програмне рішення розширення функціоналу браузерної системи управління проектами Trello на основі використання її API. Програмування проведено мовою Python. Апробація результатів підтвердила доцільність розробки при проведенні щомісячної звітності та обчислення метрик продуктивності роботи працівників.

2. Поставлені в технічному завданні задачі були розв'язані методами процедурного програмування, на основі чого реалізовано функції, що лягли в основу роботи розробленого інтелектуального модуля браузерної системи управління проектами Trello.

3. Розроблено UML-діаграму класів інтелектуального модуля, на основі якої проведено класову декомпозицію розробленого програмного модуля і впорядковано код програми відповідно до існуючої бібліотеки ru-trello.

4. Виконано агрегацію даних, відібраних з дошок, списків, карток Trello, з використанням сучасної python бібліотеки Pandas. При цьому було застосовано такі функції Pandas як зчитування, запис до файлу у форматі Microsoft Excel, групування даних, перезавантаження індексу таблиці, арифметичні додавання і множення.

Список літератури

- [1] С. В. Поперешняк, «Проблеми підготовки IT-спеціалістів», *Системи обробки інформації*. № 7 (88), с. 127-131, 2010.
 - [2] David M. Beazley, Python Essential Reference, 4th Ed. Addison-Wesley Professional, 2009. 717 p.
 - [3] Кнут Д. Э. *Искусство программирования. Т. 3. Сортировка и поиск*: пер. с англ. – 2е изд. – М.: Издательский дом «Вильямс», 2003. 832 с.
 - [4] Trello. [Електронний ресурс] – Режим доступу: <https://trello.com/>
 - [5] О. Д. Азаров, О. І. Черняк, та Л. А. Савицька, *Прикладне програмування*, Вінниця : ВНТУ, 2016. 131с.
 - [6] Т. М. Боровська, І.С. Колесник, та В.А. Северілов, *Метод оптимального агрегування в оптимізаційних задачах*Вінниця: УНІВЕРСУМ-Вінниця, 2009. – 229 с.
 - [7] Matt Telles, *Python Power! The Comprehensive Guide*, Thomson Course Technology, 2012. – 528 p.
- Стаття надійшла: 28.08.2018

Відомості про авторів

Яровий Андрій Анатолієвич, д.т.н., проф., зав. кафедри комп'ютерних наук, ВНТУ, кафедра комп'ютерних наук.

Барабан Сергій Володимирович, к.т.н., старший викладач кафедри комп'ютерних наук, ВНТУ, кафедра комп'ютерних наук, Вінниця, Хмельницьке шосе, 95.

Криночкін Роман Володимирович, к.т.н., доцент, ВНТУ, кафедра радіотехніки, r@politech.org.ua, Вінниця, Хмельницьке шосе, 95.

A. A. Yarovyi¹, S. V. Baraban¹, R. V. Krynochkin¹

INTELLECTUAL MODUL OF BROWSER MANAGEMENT SYSTEM OF IT-PROJECTS TRELLO

¹Vinnitsia National Technical University