

користувачеві не запам'ятовувати, де знаходиться навігація, а інтуїтивно щоразу переходити на потрібний розділ.

Користувач web-програми може контролювати свої доходи та витрати, також користувач може додати джерела доходу та самостійно керувати категоріями витрат. Якщо користувач бажає накопичити щось - то й для такої потреби є необхідний функціонал [4].

Сам сервіс позбавлений живої технічної підтримки, але супроводжується повною документацією і розділом FAQ, де можна подивитися відповіді на запитання, що часто ставляться. Згідно з проектуванням - додаток має бути простим, але функціональним. У цьому утримувати функціонал лише управління особистими фінансами [5].

Метою дослідження була розробка мінімального оптимального додатка для ведення обліку особистими фінансами.

За результатами дослідження було спроектовано програму з наступними функціональними модулями, що цілком задовольняє потребу користувача в управлінні власними фінансами:

- Управління джерелами особистих доходів
- Управління особистими фінансовими рахунками
- Управління категоріями витрат фінансів
- Управління цілями для накопичення фінансів
- Зведена статистика [6].

Так само в процесі дослідження дійшли висновку, що додаток необхідний на ринку, оскільки подібні послуги надають або занадто громіздкий функціонал, або недостатній для оптимальної оцінки своїх фінансів. І додаток, що розробляється, дійсно потрібно на ринку і здатний окупити себе протягом короткого часу при невисокій вартості.

Список літератури

1. Matt Stauffer. *Laravel: Up & Running: A Framework for Building Modern PHP Apps*, Edition 2. 2019. O'Reilly Media, Inc. 554 p.
2. Robert C. Martin. *Clean Coder*. 2014. mitp Verlags GmbH & Co. KG. 216 p.
3. Robert C. Martin. *Clean Architecture: Das Praxishandbuch für professionelles Softwaredesign. Regeln und Paradigmen für effiziente Software Strukturierung*. 2018. MITP-Verlags GmbH & Co. KG. 370 p.
4. Frederick P. Brooks Jr. *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition*, Edition 2. 1995. Pearson Education. 336 p.
5. Eric Freeman, Elisabeth Robson. *Head First Design Patterns: Edition 2*. 2020. O'Reilly Media, Inc. 672 p.
6. Ben Frain. *Responsive Web Design with HTML5 and CSS3*. 2012. Packt Publishing Ltd. 324

УДК 004.4

Павлюк В. В., студент
Зелінська О.В., к.т.н., доцент
доцент кафедри інформаційних технологій

АЛГОРИТМІЗАЦІЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Донецький національний університет імені Василя Стуса, м. Вінниця

Створення програмного забезпечення сьогодні є однією з передових галузей світової економіки. Даний факт абсолютно закономірний, якщо враховувати високі темпи розвитку інформаційних технологій та засобів обробки даних, які тільки удосконалюються та ускладнюються. Розробка програми – складний, багатогранний та багатоетапний процес, який вимагає від учасників високої кваліфікованості, досвіду та деякої дисциплінованості. Зокрема, для того, щоб не просто зробити відмінний програмний продукт, але зробити це ще й з мінімальними ресурсними витратами. Потрібне уважне та ретельне виконання всіх етапів розробки, визначених для конкретного проекту.

Одним із обов'язкових етапів розробки для будь-якого продукту є алгоритмізація. Алгоритм є послідовністю дій, які ведуть до бажаного результату. Алгоритмізація у процесі розробки програмного забезпечення дозволяє як структурувати роботу майбутньої програми, так і наочно показати можливі помилки і недоліки.

Мета дослідження – характеристика алгоритмізації як обов'язкового етапу розробки програми.

Для досягнення мети необхідно виконати завдання:

- охарактеризувати процес розробки програми загалом;
- проаналізувати етапи розробки програмного забезпечення;
- розглянути поняття та основні характеристики алгоритму;
- розглянути алгоритмізацію із боку обов'язкового етапу розробки програми.

Програма – це послідовність інструкцій (команд), що описує алгоритм розв'язання за допомогою комп'ютера відповідного завдання [1]. Готовими до виконання є лише програми, що містять інструкції у двійковому машинному коді – програми мовою конкретного комп'ютера (комп'ютера з процесором певної моделі чи сімейства), лише такі програми можна завантажити на згадку комп'ютера до виконання. Отже, програми у машинному коді є «переносними», їх можна виконувати лише з комп'ютерів із загальною архітектурою, системою команд, які підтримуються цими комп'ютерами, тобто однаковою машинною мовою [2].

Розробка програмного забезпечення (далі – ПЗ) – діяльність із створення нового програмного забезпечення. Як і інші традиційні інженерні дисципліни, розробка програмного забезпечення має справу з проблемами якості, вартості та надійності. Деякі програми містять мільйони рядків вихідного коду, які повинні правильно виконуватися в умовах, що змінюються. Складність ПЗ порівняна зі складністю найбільш складних із сучасних машин, таких як літаки.

Процес створення ПЗ – безліч різних видів діяльності, методів, методик і кроків, що використовуються для розробки та еволюції ПЗ та пов'язаних з ним

продуктів (проектних планів, документації, програмного коду, тестів, документації користувача тощо) [3].

Однак при формуванні загальних підходів до створення алгоритмічного та програмного забезпечення виникають труднощі, пов'язані з необхідністю досягнення їхньої наочності, структурованості, спостережливості та керованості. Ці проблеми посилюються тим, що при створенні різних систем управління зазвичай використовуються різні технології алгоритмізації та програмування. Спектр таких технологій широкий: від технологій на основі алгоритмічних мов високого рівня для промислових комп'ютерів технологій та на основі спеціалізованих мов для програмованих логічних контролерів (наприклад, функціональних та сходових схем).

Етап програмування включає вибір мови програмування та власне формалізоване (на основі пропонованих методів) ізоморфне програмування за побудованими алгоритмічними моделями. На основі запропонованого підходу розроблена програмна оболонка, що дозволяє проводити моделювання у термінах автоматів. Таким чином, використання запропонованої технології, яка отримала назву SWITCH технологія [4], дозволяє з єдиних позицій проводити алгоритмізацію та програмування при створенні систем логічного управління та знизити кількість помилок в алгоритмах та програмах. Ця технологія може бути названа також STATE-технологія або більше точно AUTOMATON-технологія, а відповідна область програмування – автоматне програмування.

Реалізація того чи іншого завдання може бути проведена безліччю способів, і алгоритмізація, будучи невід'ємною частиною процесу розробки ПЗ, дозволяє розробнику найбільш грамотно і повно оцінити ситуацію і вибрати найкраще рішення. При цьому коректно побудований алгоритм дозволяє розробникам наочно побачити структуру майбутнього проекту та проаналізувати можливі складнощі та перешкоди.

Список літератури

1. Дж. Ріхтер. *Windows для професіоналів Створення ефективних WIN32-програм з урахуванням специфіки 64-розрядної версії Windows.*
2. Кью Дж., Джеаніні М. *Об'єктно-орієнтоване програмування.*
3. Шалито А.А. *Програмна реалізація керуючих автоматів. Суднобудівна промисловість. Сер. "Автоматика та телемеханіка". 1991. Вип.13.*

УДК 004.4

Рудкевич Б. М., студент 2 курсу спеціальності 122 «Комп'ютерні науки»

Потапова Н. А., к.е.н., доцент, доцент кафедри інформаційних технологій