Тема 15. Алгоритм Луна для підрахунку контрольної цифри

Презентація

План

Про команду

Концепція

ЮМЛ

псевдокод

Код + Ексепшини

Юніт тести

Інформац ресурси

Доброго дня!

Команда – Група №3

На сьогодні наша команда складається з Сергія Перев’язка, Юрка Ничика та Юри Харченка

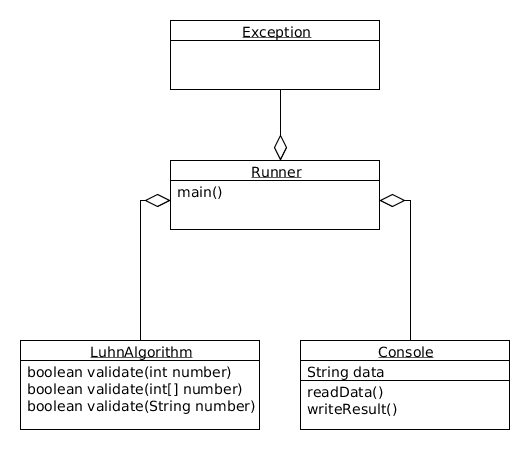
Даний алгоритм широко використовується у наступних галузях:

* Номери усіх [банківских карт](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%BD%D0%BA%D0%BE%D0%B2%D1%81%D0%BA%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%91%D0%B6%D0%BD%D0%B0%D1%8F_%D0%BA%D0%B0%D1%80%D1%82%D0%B0" \o "Банковская платёжная карта)
* Номери дисконтних карт
* Коди соц страхування
* [IMEI](https://ru.wikipedia.org/wiki/IMEI)-коди.

Алгоритм Луна дозволяє виявити помилково введені дані, а саме, номер карти, яка потім використовуються для різноманітних операцій. Використовується для номерів різної довжини, парної та не парної кількості цифр.

Концепція наближена до реальної ситуації, а саме, користувач виконує введення номера платіжної картки з клавіатури(в консоль), та отримує відповідь у вигляді інформаційного повідомлення про коректність даних згідно алгоритму.

*Огляд діаграми типу UML*



Клас LuhnAlgorithm містить код алгоритму для різних типів вхідних даних – числа, масив, рядок.

Клас Console описує методи вводу та виводу даних від/для користувача.

Створений клас Runner містить метод main(), де і виконується запуск нашої програми. Зв’язок класів Runner – LuhnAlgorithm та Runner – Console відповідає типу Агрегація, що в даному випадку означає, що клас Runner “has a” методи класів LuhnAlgorithm та Console.

Ну і, звичайно, клас Runner при запуску програми передбачає обробку певних виключних ситуацій, вказаних у класі Exception.

*Опис Алгориту Луна*

Наприклад, користувач ввів номер карти:

4 5 6 1 2 6 1 2 1 2 3 4 5 4 6 4

Якщо кількість цифр парна(наш випадок), то виконується наступна перевірка, починаючи з першої цифри:

Якщо X\*2 > 9, то від результату віднімаємо 9 та записуємо новий результат на місце перевіряємої цифри , якщо умова не виконується, то записуємо результат X\*2.

4 5 6 1 2 6 1 2 1 2 3 4 5 4 6 4

8 3 4 2 2 6 1 3

Далі номер має наступний вигляд:

8 5 3 1 4 6 2 2 2 2 6 4 1 4 3 4

Потім необхідно скласти всі числа. Результат (тут 57) повинен бути кратним 10.

Отже даний номер карти не коректний.

Як варіант, користувач міг просто помилково вказати останню контрольну цифру.

Тобто, остання цифра є контрольною, так як являється результатом послідовності попередніх 15 цифр.

Якщо її змінити на 7, то загальна сума буде 60, що буде відповідати критеріям алгоритму.

*Код програми та ексепшини – DEMO*

*Unit тести – DEMO*

Інформаційні ресурси

* Основний підхід, що вико використовувався при написанні даної програми: <https://lurkmore.to/KISS>
* [U.S. Patent 2 950 048](http://www.google.com/patents/US2950048) *Computer for Verifying Numbers*, Hans P. Luhn, August 23, 1960.
* <https://en.wikipedia.org/wiki/Luhn_algorithm>