

VISION

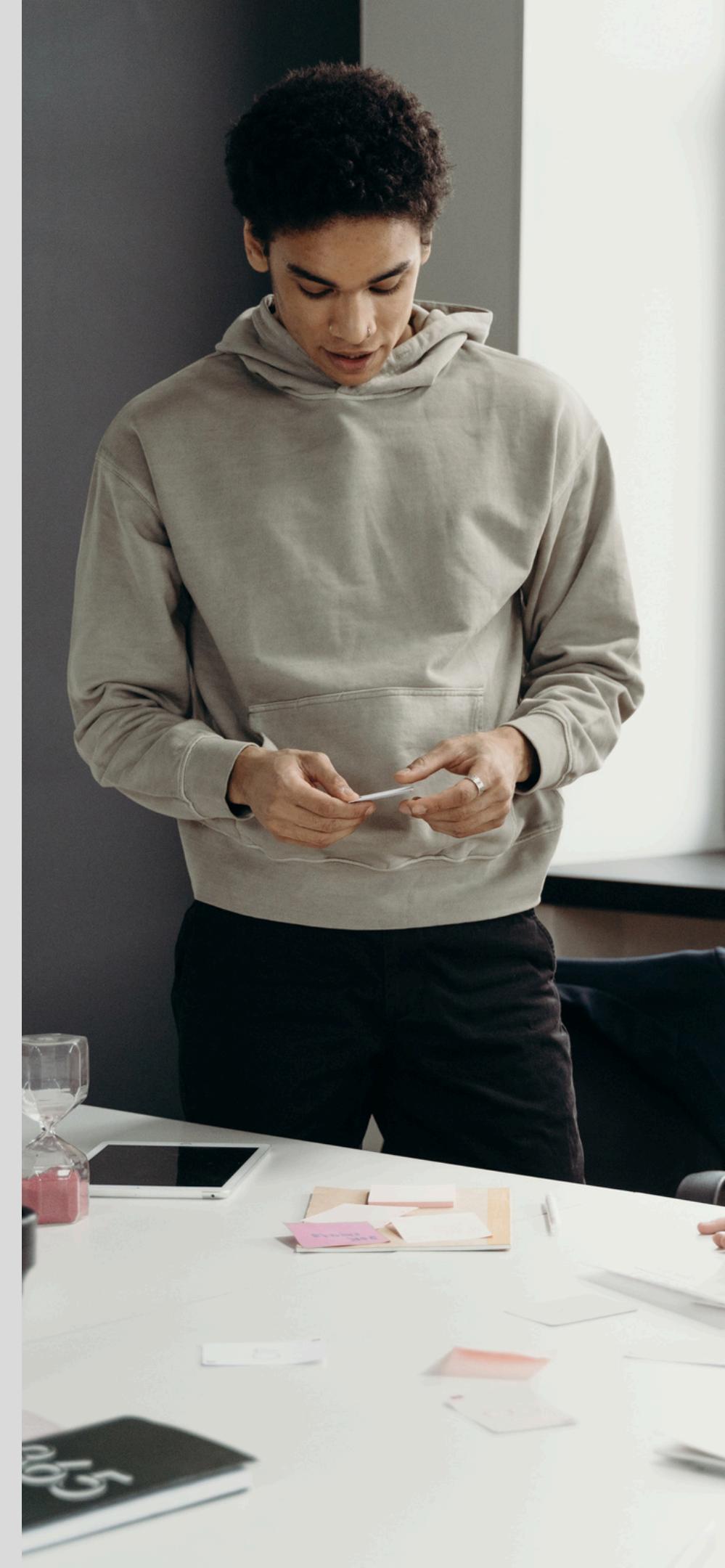
DOT



Agenda



- Introduction
- OOPS concepts used
- System Overview
- Results and system interface
- Conclusion
- References



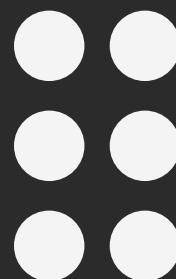
INTRODUCTION

Braille is a tactile writing system used by people who are visually impaired. It was invented by Louis Braille in 1824. The system uses patterns of raised dots arranged in cells of up to six dots in a 3×2 configuration. Each pattern represents a character, numeral, or punctuation mark.

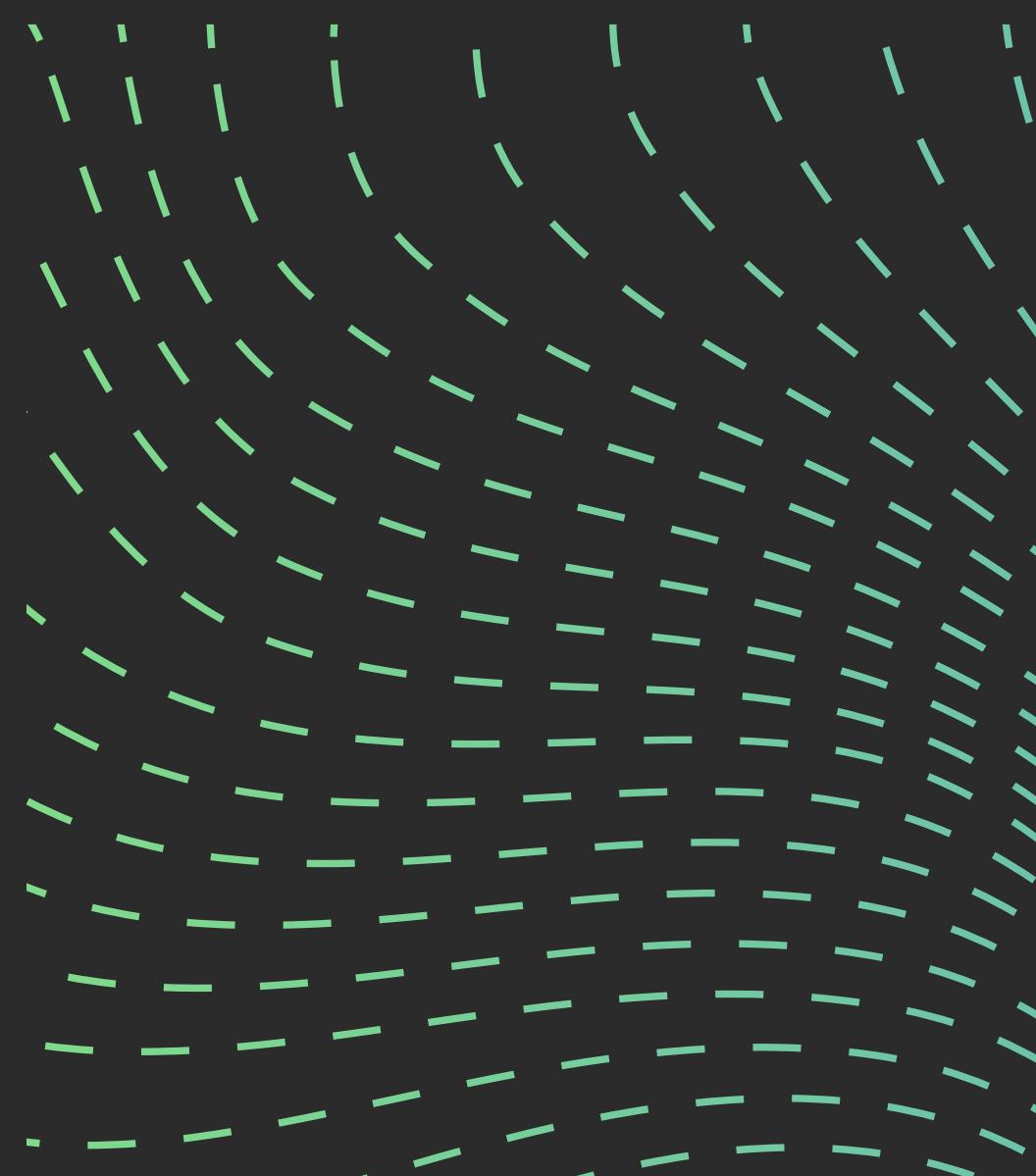
Digital Braille Representation

In digital systems, Braille dots are often represented using binary notation:

Dot Positions: Standard Numbering:



14
25
36



OOPs concept used



- Abstraction : Hiding implementation details and showing only essential features
- Encapsulation : Bundling data and methods that operate on that data within a single unit
- Inheritance : Creating new classes based on existing classes
- Polymorphism : One interface, multiple implementations
- Exception Handling : Handling runtime errors gracefully

USER INTERFACE LAYER

VisionDotsFX (JavaFX)

- Text input area
- Convert button
- Braille output display



BUSINESS LOGIC LAYER

TextToBraille (Processor)

- Handles number/capital modes
- Formats output into 3 rows
- Coordinates conversion process



DATA ACCESS LAYER

BrailleMap (Data Storage)

- Stores character-to-Braille mappings
- Inherited by TextToBraille



INTERFACE LAYER

BrailleConverter (Interface)

- Defines convert() method contract
- Ensures consistent implementation

SYSTEM OVERVIEW

1. UI Layer

- Text input
- Convert button
- Braille output display

2. Business Logic Layer

- Handles capital & number modes
- Converts text to Braille
- Formats output into 3 rows

3. Data Layer

- Stores character-Braille mappings
- Uses HashMap structure

4. Interface Layer

- Defines convert() method
- Ensures consistent conversion

Results and System interface

- Braille Encoding Scheme

- Letter Mapping Examples:

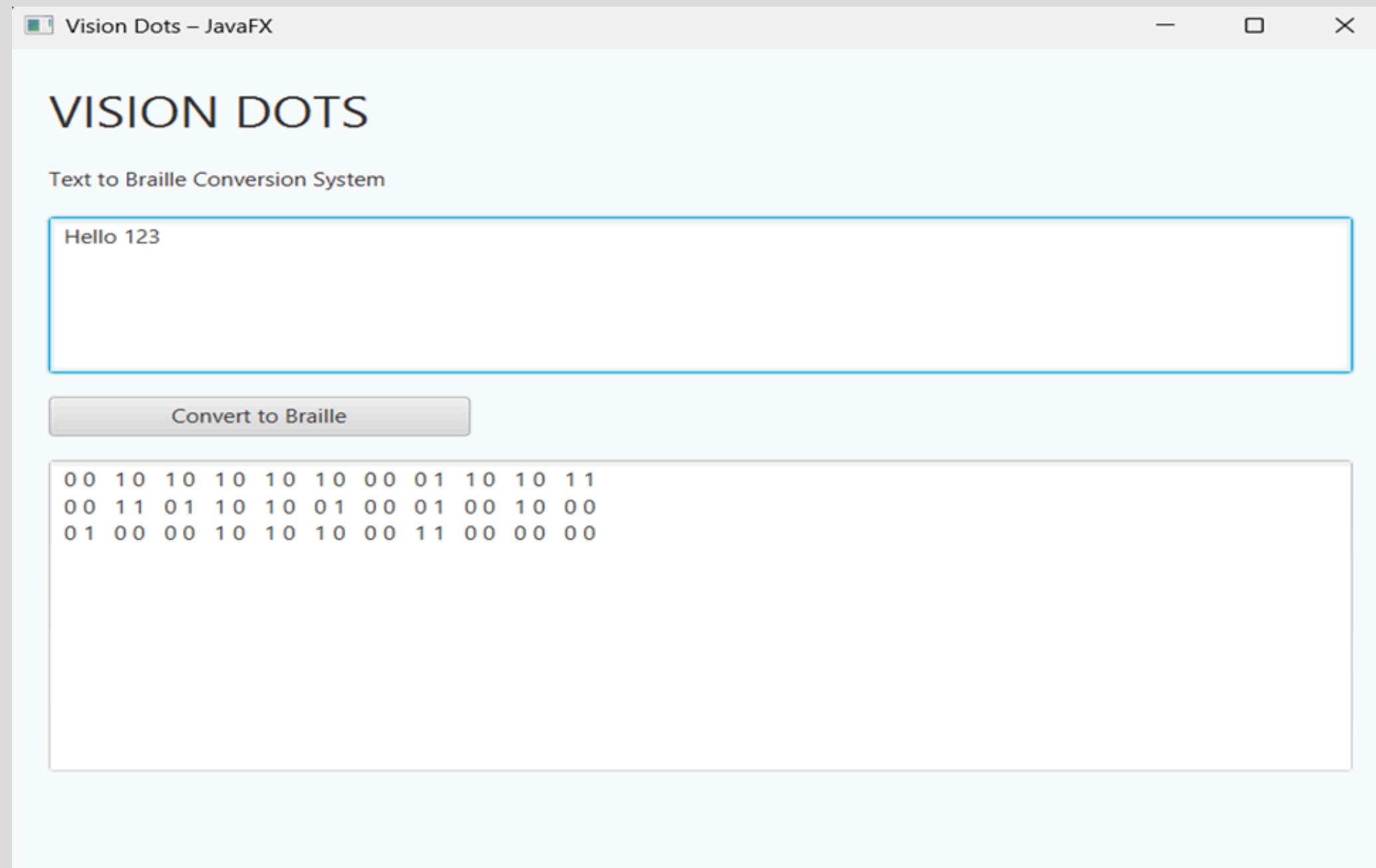
- a = 100000 (● ○ ○ ○ ○)
- b = 110000 (● ● ○ ○ ○)
- c = 100100 (● ○ ○ ● ○ ○)

- Number Encoding (with number sign prefix):

- 1 = 001111 100000
- 2 = 001111 110000

- Special Signs:

- Capital: 000001
- Number: 001111
- Space: 000000



>>>

Conclusion

The Vision Dot project successfully delivers a comprehensive Braille conversion system that transforms English text into Braille patterns. This project effectively applies core object-oriented programming principles, showcasing modular design and scalability. The inclusion of a user-friendly graphical interface enhances accessibility, while robust error handling ensures smooth operation.

While currently limited to English characters and Grade 1 Braille without audio or tactile feedback, Vision Dot lays a strong foundation for educational purposes and Braille learning. Future improvements aim to expand language support, introduce Grade 2 Braille contractions, add audio feedback for visually impaired users, enable export features, and develop a mobile application to broaden usability.

Overall, Vision Dot demonstrates both technical strength and educational value, contributing meaningfully to assistive technology solutions.

References

- Official Documentation

Oracle Java Documentation: <https://docs.oracle.com/javase/>

JavaFX Documentation: <https://openjfx.io/>

Unicode Braille Patterns: <https://unicode.org/charts/PDF/U2800.pdf>

- Technical Resources

Braille Authority of North America. (2020). Braille Codes and Formats

World Blind Union. (2019). International Guide to Braille Notation

Java Design Patterns: <https://java-design-patterns.com/>

- Academic References

Deitel, P. J., & Deitel, H. M. (2017). Java: How to Program

Bloch, J. (2018). Effective Java

Horstmann, C. S. (2019). Core Java Volume I: Fundamentals

- Online Resources

W3Schools Java Tutorial: Basic Java concepts

THANK YOU

By

MANISHA CS

MANISH NAYAKA R

MANIKANTA KARADIGUDDA

MANJU KS