CLAUDE AI

 random: Module for generating random selections

 string: Provides constants for string operations (lowercase, uppercase, digits, punctuation)

 re: Regular expression module for pattern matching and validation

 Creates a class to manage security code generation and validation

 __init__: Constructor method called when class is instantiated

 self.min_length: Minimum allowed code length (6 characters)

 self.max_length: Maximum allowed code length (12 characters)

 complexity_levels: Dictionary defining minimum character count for each complexity level

 Method to generate security codes

 complexity parameter defaults to 'medium'

 random.randint(): Generates random integer between minimum complexity length and max length

 Ensures variability in generated code length

 Creates character sets from string module

 ascii_lowercase: All lowercase letters (a-z)

 ascii_uppercase: All uppercase letters (A-Z)

 digits: Numeric characters (0-9)

 symbols: Punctuation characters

 Dynamically builds character set based on selected complexity

 'low': Only lowercase and digits

 'medium': Lowercase, uppercase, and digits

 'high': All character types including symbols

 Uses list comprehension to generate random code

 random.choice(): Selects random character from character set

 ''.join(): Converts list of characters to single string

 for _ in range(length): Repeats selection process for specified length

 Checks code length against minimum requirement

 Returns False and error message if too short

 Returns tuple with validation status and feedback

- Dictionary defining regex patterns for different character types

- r'[a-z]': Matches any lowercase letter

- r'[A-Z]': Matches any uppercase letter

- r'\d': Matches any digit

- r'[!@#$%^&*(),.?":{}|<>]': Matches specific symbol characters

- List comprehension to find missing character types

- checks.items(): Iterates through check dictionary

- re.search(): Searches for pattern in code

- Builds list of character types not present in code

- f any checks failed, generate feedback message

- ', '.join(): Converts list of failed checks to comma-separated string

- Returns validation result and feedback

- Creates instance of SecurityCodeManager

- Provides interactive menu for code generation and validation

- Infinite loop for menu-driven interface

- Displays options to user

- Captures user's menu selection

- Option 1: Generate security code

- Prompts for complexity level

- Defaults to 'medium' if invalid input

- Calls code generation method

- Prints generated code

- Option 2: Validate security code

- Prompts for code input

- Calls validation method

- Displays validation result and feedback