

| Nr | Obszar   | Wymaganie              | KOD  |                          | Przyznane pkt | Pkt max |
|----|----------|------------------------|--|--------------------------|---------------|---------|
| 1  | UI       | JEST                   | <pre>while True:     print("Choose an option:")     print("1. Learn Morse Code")     print("2. Encoder")     print("3. Decoder")     print("4. Exit")     choice = input("Enter your choice (1-4): ")</pre>  | <input type="checkbox"/> |               |         |
|    |          | Wprowadzanie danych    | <pre>choice = input("Enter your choice (1-4): ")</pre>   | <input type="checkbox"/> |               | 2       |
|    |          | Wyświetlanie danych    | <pre>print("Morse Code Chart:") for letter, code in morseCode.items():     print(f'{letter}: {code}')</pre>  | <input type="checkbox"/> |               | 2       |
|    |          | Zmiana danych          | <pre>if self.level &lt; 6:     self.level += 1     print(f'Level up! Now you are at level {self.level}.')     self.writeFile()</pre>   | <input type="checkbox"/> |               | 2       |
|    |          | Wyszukiwanie danych    | <pre>if char.upper() in morseCode:     encoded_text.append(morseCode[char.upper()])</pre>  | <input type="checkbox"/> |               | 2       |
|    |          | Przedstawienie wyników | <pre>print(f'Level up! Now you are at level {self.level}.')</pre>  | <input type="checkbox"/> |               | 2       |
| 2  | Podstawy | Zmienne                | <pre>choice = input("Enter your choice (1-4): ")</pre>   | <input type="checkbox"/> |               | 2       |
|    |          | typy danych            | <pre>morseCode = {     'A': '-.-',     'B': '-...',     'C': '-.-.-',     'D': '-..',     'E': '.',     'F': '-.-.-',     'G': '--.',     'H': '....',     'I': '..',     'J': '-.-.-',     'K': '-.-',     'L': '-.-.-',     'M': '--',     'N': '-.-',     'O': '---',     'P': '-.-.-',     'Q': '-.-.-',     'R': '-.-',     'S': '...',     'T': '-',     'U': '-.-',     'V': '...-',     'W': '-.-.-',     'X': '-.-.-',     'Y': '-.-.-',     'Z': '--.-',     '0': '-----',     '1': '-.-.-.-',</pre> | <input type="checkbox"/> |               | 2       |

|  |            |  |                          |  |     |
|--|------------|--|--------------------------|--|-----|
|  |            | <pre> '2': '..--', '3': '...--', '4': '....--', '5': '.....', '6': '.....', '7': '....', '8': '....', '9': '----', ':': '/', '.': '..--', ';': '....', '?': '....', '!': '....', }  words = {     'HELLO', 'WORLD',     'PYTHON', 'MORSE', 'CODE',     'LEARN', 'PROGRAMMING',     'ALGORITHM', 'DATA',     'STRUCTURE', 'FUNCTION',     'VARIABLE', 'CLASS',     'OBJECT', 'METHOD',     'LIBRARY', 'MODULE',     'PACKAGE', 'SCRIPT',     'DEBUG',     'TEST', 'EXCEPTION',     'IMPORT', 'EXPORT',     'ENCODE', 'DECODE',     'STRING',     'INTEGER', 'FLOAT',     'BOOLEAN', 'LIST', 'TUPLE',     'SET', 'DICTIONARY',     'ARRAY', 'LOOP',     'CONDITION', 'IF', 'ELSE',     'ELIF', 'FOR', 'WHILE',     'SWITCH', 'CASE', 'BREAK',     'CONTINUE', 'RETURN',     'YIELD', 'ASSERT',     'TRY', 'EXCEPT', 'FINALLY',     'RAISE', 'WITH', 'AS',     'IMPORTS', 'EXPORTS',     'ENCODING', 'DECODING',     'STRINGIFY', 'PARSE',     'COMPILE', 'EXECUTE', } </pre> |                          |  |     |
|  | komentarze | # print(f"Data successfully written to {self.file_path}.")   | <input type="checkbox"/> |  | 1   |
|  | operatory  | <pre> if self.level == 1 or self.level == 2:     for letter in morseCode.keys():  self.toLearn.append(letter)     elif self.level == 3 or self.level == 4:         self.toLearn = words     elif self.level == 5 or self.level == 6:         self.toLearn = sentences </pre>   | <input type="checkbox"/> |  | 1,5 |
|  | Instrukcje | <pre> if self.level == 1 or self.level == 2: </pre>  | <input type="checkbox"/> |  |     |

|  |                            |  |                          |  |   |
|--|----------------------------|--|--------------------------|--|---|
|  | warunkowe (if, elif, else) | <pre> for letter in morseCode.keys():  self.toLearn.append(letter)     elif self.level == 3 or self.level == 4:         self.toLearn = words     elif self.level == 5 or self.level == 6:         self.toLearn = sentences </pre>  |                          |  | 3 |
|  | Instrukcje iteracyjne      |  |                          |  |   |
|  | for                        | <pre> for letter, code in morseCode.items():     print(f'{letter}: {code}') </pre>   | <input type="checkbox"/> |  | 2 |
|  | while                      | <pre> while True:     print("Choose an option:")     print("1. Learn Morse Code")     print("2. Encoder")     print("3. Decoder")     print("4. Show morse code chart")     print("5. Exit")     choice = input("Enter your choice (1-5): ")      if choice == '1':         print("Starting Morse Code learning...")         learning = MorseLearning()         learning.learn()     elif choice == '2':         print("Enter text to encode into Morse code:")         text = input()         encoder = Encoder()         encoded_text = encoder.action(text)         print(f'Encoded Morse Code: {encoded_text}')     elif choice == '3':         print("Enter Morse code to decode (separate letters with spaces):")         morse_code = input()         decoder = Decoder()         decoded_text = decoder.action(morse_code)         print(f'Decoded Text: {decoded_text}')     elif choice == '4':         print("Morse Code Chart:")         for letter, code in morseCode.items():             print(f'{letter}: {code}')     elif choice == '5':         print("Exiting the </pre> | <input type="checkbox"/> |  | 2 |

|   |           |   |   |                          |  |     |
|---|-----------|---|---|--------------------------|--|-----|
|   |           |   | <pre> program.")     break     else:         print("Invalid choice. Please try again.") </pre>  |                          |  |     |
|   |           | Operacje wejścia (input)                        | <pre> choice = input("Enter your choice (1-5): ") </pre>  | <input type="checkbox"/> |  | 1,5 |
|   |           | Operacje wyjścia (print)                        | <pre> print("Starting Morse Code learning...") </pre>   | <input type="checkbox"/> |  | 1,5 |
|   |           | Funkcje z parametrami i wartościami zwracanymi  | <pre> def action(self, text):     encoded_text = []     for char in text:         try:             if char.upper() in morseCode:  encoded_text.append(morseC ode[char.upper()])         else:             raise MorseCodeNotFoundError(ch ar)         except MorseCodeNotFoundError as e:             print(e.message)     return ' '.join(encoded_text) </pre> | <input type="checkbox"/> |  | 2   |
|   |           | Funkcje rekurencyjne                            |   | <input type="checkbox"/> |  | 3   |
|   |           | Funkcje przyjmujące inne funkcje jako argumenty |   | <input type="checkbox"/> |  | 3   |
|   |           | Dekoratory                                      |   | <input type="checkbox"/> |  | 1,5 |
| 3 | Kontenery | Użycie listy                                    | <pre> encoded_text = [] </pre>  | <input type="checkbox"/> |  | 2   |
|   |           | Użycie słownika                                 | <pre> morseCode = { 'A': '.-.', 'B': '-...', 'C': '-.-.', 'D': '-..', 'E': '.', 'F': '..-.', 'G': '--.', 'H': '....', 'I': '..', 'J': '-.-.-', 'K': '-.-', 'L': '-..', 'M': '--', 'N': '-.', 'O': '---', 'P': '-.-.', 'Q': '--.-', 'R': '-.-', 'S': '...', 'T': '.', 'U': '..-', 'V': '...', 'W': '.-.-', </pre>  | <input type="checkbox"/> |  | 2   |

|  |  |               |  |                          |  |     |
|--|--|---------------|--|--------------------------|--|-----|
|  |  |               | <pre>'X': '-.-', 'Y': '-.-', 'Z': '-.-', '0': '-----', '1': '-----', '2': '-----', '3': '-----', '4': '-----', '5': '-----', '6': '-----', '7': '-----', '8': '-----', '9': '-----', ':': '/', '.': '-----', ';': '-----', '?': '-----', '!': '-----', }</pre>   |                          |  |     |
|  |  | Użycie zbioru | <pre>words = {     'HELLO', 'WORLD',     'PYTHON', 'MORSE', 'CODE',     'LEARN', 'PROGRAMMING',     'ALGORITHM', 'DATA',     'STRUCTURE', 'FUNCTION',     'VARIABLE', 'CLASS',     'OBJECT', 'METHOD',     'LIBRARY', 'MODULE',     'PACKAGE', 'SCRIPT',     'DEBUG',     'TEST', 'EXCEPTION',     'IMPORT', 'EXPORT',     'ENCODE', 'DECODE',     'STRING',     'INTEGER', 'FLOAT',     'BOOLEAN', 'LIST', 'TUPLE',     'SET', 'DICTIONARY',     'ARRAY', 'LOOP',     'CONDITION', 'IF', 'ELSE',     'ELIF', 'FOR', 'WHILE',     'SWITCH', 'CASE', 'BREAK',     'CONTINUE', 'RETURN',     'YIELD', 'ASSERT',     'TRY', 'EXCEPT', 'FINALLY',     'RAISE', 'WITH', 'AS',     'IMPORTS', 'EXPORTS',     'ENCODING', 'DECODING',     'STRINGIFY', 'PARSE',     'COMPILE', 'EXECUTE', }</pre> | <input type="checkbox"/> |  | 1,5 |
|  |  | Użycie krotki | <pre>sentences = (     'HELLO WORLD', 'PYTHON     IS FUN', 'LEARN MORSE     CODE', 'PROGRAMMING IS     GREAT',     'DATA STRUCTURES ARE     IMPORTANT', 'FUNCTIONS     AND METHODS', 'OBJECT     ORIENTED PROGRAMMING',     'ALGORITHMS AND     COMPLEXITY', 'DEBUGGING     AND TESTING', 'EXCEPTIONS     AND ERROR HANDLING',</pre>   | <input type="checkbox"/> |  | 1,5 |

|   |                      |                                 |  |                          |  |     |
|---|----------------------|---------------------------------|--|--------------------------|--|-----|
|   |                      |                                 | 'IMPORTING MODULES',<br>'EXPORTING DATA',<br>'ENCODING, DECODING<br>STRINGS',<br>'LISTS AND DICTIONARIES<br>IN PYTHON', 'LOOPS AND<br>CONDITIONS IN<br>PROGRAMMING'<br>'S.O.S', 'DO YOU LIKE<br>PYTHON?', 'MORSE CODE IS<br>COOL!'<br>)  |                          |  |     |
| 4 | Przestrzenie<br>nazw | Zastosowano<br>zmienne lokalne  | <pre> for char in text:     try:         if char.upper() in morseCode:  encoded_text.append(morseC ode[char.upper()])     else:         raise MorseCodeNotFoundError(ch ar)     except MorseCodeNotFoundError as e:         print(e.message) </pre>  | <input type="checkbox"/> |  | 1,5 |
|   |                      | Zastosowano<br>zmienne globalne | <pre> sentences = (     'HELLO WORLD',     'PYTHON IS FUN', 'LEARN MORSE CODE',     'PROGRAMMING IS GREAT',     'DATA STRUCTURES ARE IMPORTANT',     'FUNCTIONS AND METHODS', 'OBJECT ORIENTED PROGRAMMING',     'ALGORITHMS AND COMPLEXITY',     'DEBUGGING AND TESTING', 'EXCEPTIONS AND ERROR HANDLING',     'IMPORTING MODULES',     'EXPORTING DATA',     'ENCODING, DECODING STRINGS',     'LISTS AND DICTIONARIES IN PYTHON', 'LOOPS AND CONDITIONS IN PROGRAMMING'     'S.O.S', 'DO YOU LIKE PYTHON?', 'MORSE CODE IS COOL!', ) </pre> | <input type="checkbox"/> |  | 1,5 |
|   |                      | Zastosowano                     | encoded_text = []  | <input type="checkbox"/> |  |     |

|    |                  | zakresy funkcji  | <pre> for char in text:     try:         if char.upper() in morseCode:  encoded_text.append(morseC ode[char.upper()])     else:         raise MorseCodeNotFoundError(ch ar)     except MorseCodeNotFoundError as e:         print(e.message)     return ' '.join(encoded_text) </pre>   |                          |               | 1,5     |
|----|------------------|--|---|--------------------------|---------------|---------|
|    |                  | Zastosowano zakresy klas                                 | <pre> def __init__(self, file_path):     self.file_path = file_path </pre>  | <input type="checkbox"/> |               | 1,5     |
| 5  | Moduły i pakiety | Projekt podzielony na moduły (import, __init__)          | <pre> from .coders import * from .errors import * from .utils import * from .fileManager import * from .learning import * </pre>  | <input type="checkbox"/> |               | 2       |
| Nr | Obszar           | Wymaganie  | KOD   |                          | Przyznane pkt | Pkt max |
|    |                  | Własne pakiety/funkcje pomocnicze w osobnych plikach .py | <pre> from .coders import * from .errors import * from .utils import * from .fileManager import * from .learning import * </pre>  | <input type="checkbox"/> |               | 2       |
| 6  | Obsługa błędów   | Obsługa wyjątków (try, except, finally)                  | <pre> try:     with open(self.file_path, 'r') as file:         data = json.load(file)     return data except FileNotFoundError:     print(f'File {self.file_path} not found.")     return None except json.JSONDecodeError:     print(f'Error decoding JSON from file {self.file_path}.")     return None except Exception as e:     print(f'An error occurred while reading the file: {e}')     return None </pre> | <input type="checkbox"/> |               | 2       |
|    |                  | Użycie assert do testów i walidacji                      |   | <input type="checkbox"/> |               | 1,5     |
| 7  | Łańcuchy         | Operacje na  | morse_words =   | <input type="checkbox"/> |               |         |

|     |                         |   |   |                          |  |     |
|-----|-------------------------|---|---|--------------------------|--|-----|
|     | znaków                  | stringach (m.in. formatowanie, dzielenie, wyszukiwanie) | morse_code.split(' ')   | <input type="checkbox"/> |  | 2   |
| 8   | Obsługa plików          | Odczyt z plików .txt, .csv, .json, .xml (min. 1)        | with<br>open(self.file_path, 'r') as file:<br>data =<br>json.load(file)<br>return data                              | <input type="checkbox"/> |  | 2   |
|     |                         | Zapis do plików .txt, .csv, .json, .xml (min. 1)        | with<br>open(self.file_path, 'w') as file:<br>json.dump(data, file,<br>indent=4)                                    | <input type="checkbox"/> |  | 2   |
| 9   | OOP                     | Klasy   | class Coder:  | <input type="checkbox"/> |  | 2   |
|     |                         | Metody  | class Coder:<br>def action(self, text):<br>raise<br>NotImplementedError("Subclasses should implement this method.") | <input type="checkbox"/> |  | 2   |
|     |                         | Konstruktory  | class FileManager:<br>def __init__(self, file_path):<br>self.file_path = file_path                                  | <input type="checkbox"/> |  | 2   |
|     |                         | Dziedziczenie   | class Encoder(Coder):   | <input type="checkbox"/> |  | 2   |
| 10  | Programowanie funkcyjne | map   |   | <input type="checkbox"/> |  | 1,5 |
|     |                         | filter  | tmpList = filter(lambda letter:<br>letter != item, self.toLearn)  | <input type="checkbox"/> |  | 1,5 |
|     |                         | lambda  | tmpList = filter(lambda letter:<br>letter != item, self.toLearn)  | <input type="checkbox"/> |  | 1,5 |
|     |                         | reduce  |   | <input type="checkbox"/> |  | 1,5 |
| 11  | Wizualizacja danych     | Wygenerowano wykres (np. matplotlib, seaborn)           |   | <input type="checkbox"/> |  | 2   |
|     |                         | Zapisano wykres do pliku graficznego (.png lub .jpg)    |   | <input type="checkbox"/> |  | 1,5 |
| T12 | Testowanie              | Testy jednostkowe (assert, unittest, pytest)            | self.assertEqual(self.decoder.action('.... .-... ..- / .-- --- .-. ...- / ..'), 'HELLO WORLD I')                    | <input type="checkbox"/> |  | 1,5 |
|     |                         | Testy funkcjonalne                                      |   | <input type="checkbox"/> |  | 1,5 |
|     |                         | Testy Integracyjne                                      |   | <input type="checkbox"/> |  | 1,5 |
|     |                         | Testy graniczne / błędne dane                           | self.assertEqual(self.decoder.action(""), "")   | <input type="checkbox"/> |  | 1,5 |
|     |                         | Testy wydajności (np. czas wykonania, timeit)           |   | <input type="checkbox"/> |  | 1,5 |
|     |                         | Testy pamięci memory_profiler                           |   | <input type="checkbox"/> |  | 1,5 |
|     |                         | Test jakości kodu (flake8, pylint)                      |   | <input type="checkbox"/> |  | 1,5 |
| 13  | Wersjonowanie           | Repozytorium GIT  | <a href="https://github.com/kyzelm/js-python-project">https://github.com/kyzelm/js-python-project</a>               | <input type="checkbox"/> |  | 1   |
|     |                         | Historia commitów                                       | commit<br>7a6d112e3d9f97bcfbf00296a597  | <input type="checkbox"/> |  | 1   |



|    |              |   | f26fa5ba7b65 (HEAD -> main,<br>origin/main)<br>Author: Damian Osiński<br><osinskidamian03@gmail.com><br>Date: Mon Jun 23 16:49:41<br>2025 +0200<br><br>Fixes<br><br>commit<br>785f39ac6adb97f31a042b52831b<br>7acd294a8405<br>Author: Damian Osiński<br><osinskidamian03@gmail.com><br>Date: Mon Jun 23 02:47:59<br>2025 +0200<br><br>Tests  |                          |                  |            |
|----|--------------|---|--|--------------------------|------------------|------------|
| Nr | Obszar       | Wymaganie   | KOD  |                          | Przyznane<br>pkt | Pkt<br>max |
|    |              | Link do GitHub                                    | <a href="https://github.com/kyzelm/js-python-project">https://github.com/kyzelm/js-python-project</a>  | <input type="checkbox"/> |                  | 1          |
|    |              | Opis commitów                                     | commit<br>9f7afd6f424b53a2afad4477656de<br>3a57ce4dee9<br>Author: Damian Osiński<br><osinskidamian03@gmail.com><br>Date: Mon Jun 23 01:35:45<br>2025 +0200<br><br>Add README and<br>requirements files   | <input type="checkbox"/> |                  | 1          |
| 14 | Dokumentacja | Plik README.md<br>(cel, autorzy,<br>uruchamianie) | # Naucz się kodu Morse'a<br><br>Projekt zawiera, prosty<br>program w Pythonie, który<br>pozwala na naukę kodu<br>Morse'a. Program jest<br>interaktywny i umożliwia<br>użytkownikowi naukę poprzez<br>ćwiczenia oraz enkodowanie i<br>dekodowanie wiadomości w<br>kodzie Morse'a.<br><br>## Instalacja<br>Aby zainstalować wymagane<br>biblioteki, uruchom<br>następujące polecenie:<br><br><pre>```bash pip install -r requirements.txt ```</pre><br>## Użycie<br>Aby uruchomić program, użyj<br>następującego polecenia:<br><br><pre>```bash python main.py ```</pre> | <input type="checkbox"/> |                  | 1,5        |

|  |  |  |   |                          |  |   |
|--|--|--|---|--------------------------|--|---|
|  |  |  | ## Twórcy<br>- Damian Osiński<br>- Kacper Nyga  |                          |  |   |
|  | Przykładowe dane wejściowe i wyjściowe |  | C:\Users\damio\Dev\js-projekt\venv\Scripts\python.exe C:\Users\damio\Dev\js-projekt\main.py<br>Choose an option:<br>1. Learn Morse Code<br>2. Encoder<br>3. Decoder<br>4. Show morse code chart<br>5. Exit<br>Enter your choice (1-5): 1<br>Starting Morse Code learning...<br>Translate to Morse Code:<br>G<br>Your answer:<br>Incorrect! The correct answer is:<br>--.<br>Translate to Morse Code:<br>I<br>Your answer: ..<br>Correct!<br>Translate to Morse Code:<br>Q<br>Your answer: exit<br>Exiting the learning session.<br>Choose an option:<br>1. Learn Morse Code<br>2. Encoder<br>3. Decoder<br>4. Show morse code chart<br>5. Exit<br>Enter your choice (1-5): 2<br>Enter text to encode into Morse code:<br>Ala ma kota<br>Encoded Morse Code: .- .-. .- / -- .- / -.- ---- .-<br>Choose an option:<br>1. Learn Morse Code<br>2. Encoder<br>3. Decoder<br>4. Show morse code chart<br>5. Exit<br>Enter your choice (1-5): 5<br>Exiting the program.<br><br>Process finished with exit code 0 | <input type="checkbox"/> |  | 2 |
|  | Diagram klas lub struktura modułów     |  |   | <input type="checkbox"/> |  | 2 |
|  | SUMA                                   |  |   |                          |  |   |