PRZYPADKI TESTOWE

1. UtilityTest

package pl.sudokusolver.solver.utility

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.1 | | | |
| Tytuł | UsedInRow0TestTrue | **Oczekiwany rezultat** | Każde użycie funkcji usedInRow dla zerowego wiersza zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w zerowym wierszu | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.2 | | | |
| Tytuł | UsedInRowTestIndex-  OutOfBoundException | **Oczekiwany rezultat** | Rzucenie IndexOutOfBoundException |
| Opis | Sprawdzenie czy dla wierszy nie występujących w macierzy funkcja rzuci wyjątek | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.3 | | | |
| Tytuł | UsedInRow0TestFalse | **Oczekiwany rezultat** | Każde użycie funkcji usedInRow dla zerowego wiersza zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje w zerowym wierszu | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.4 | | | |
| Tytuł | UsedInRowTestTrue | **Oczekiwany rezultat** | Każde użycie funkcji usedInRow dla danego wiersza zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w danym wierszu | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.5 | | | |
| Tytuł | UsedInRowTestFalse | **Oczekiwany rezultat** | Każde użycie funkcji usedInRow dla danego wiersza zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje w danym wierszu | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.6 | | | |
| Tytuł | UsedInCol0TestTrue | **Oczekiwany rezultat** | Każde użycie funkcji usedInRow dla zerowej kolumny zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w zerowej kolumnie | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.7 | | | |
| Tytuł | UsedInColTestIndex-  OutOfBoundException | **Oczekiwany rezultat** | Rzucenie IndexOutOfBoundException |
| Opis | Sprawdzenie czy dla kolumn nie występujących w macierzy funkcja rzuci wyjątek | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.8 | | | |
| Tytuł | UsedInCol0TestFalse | **Oczekiwany rezultat** | Każde użycie funkcji usedInCol dla zerowej kolumny zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje w zerowej kolumnie | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.9 | | | |
| Tytuł | UsedInColTestTrue | **Oczekiwany rezultat** | Każde użycie funkcji usedInCol dla danej kolumny zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w danej kolumnie | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.10 | | | |
| Tytuł | UsedInColTestFalse | **Oczekiwany rezultat** | Każde użycie funkcji usedInCol dla danej kolumny zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje  w kolumnie | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.11 | | | |
| Tytuł | UsedInBox1TestTrue | **Oczekiwany rezultat** | Funkcja usedInBox zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w bloku 3x3, gdzie górnym lewym rogiem jest row=0,col=0 | **Poprawność wykonania** | Ok |
| Warunek wstępny | brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.12 | | | |
| Tytuł | UsedInBoxTestIndex-  OutOfBoundException | **Oczekiwany rezultat** | Rzucenie IndexOutOfBoundException |
| Opis | Sprawdzenie czy funkcja rzuci wyjątek, jeżeli nasz blok 3x3 będzie wychodził poza macierz | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.13 | | | |
| Tytuł | UsedInBox1TestFalse | **Oczekiwany rezultat** | Funkcja usedInBox zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje w bloku 3x3, gdzie górnym lewym rogiem jest row=0,col=0 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.14 | | | |
| Tytuł | UsedInBoxTestTrue | **Oczekiwany rezultat** | Każde użycie funkcji usedInBox dla danego danego bloku zwraca true |
| Opis | Sprawdzenie czy dana liczba występuje w danym bloku 3x3 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.15 | | | |
| Tytuł | UsedInBoxTestFalse | **Oczekiwany rezultat** | Każde użycie funkcji usedInBox dla danego bloku zwraca false |
| Opis | Sprawdzenie czy dana liczba nie występuje w danym bloku 3x3 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.16 | | | |
| Tytuł | CanPlaceDigitTestTrue | **Oczekiwany rezultat** | Dla zadanych wartości funkcja CanPlaceDigit zwraca true |
| Opis | Sprawdzenie czy dana liczba nie występuje w danej kolumnie, wierszu i bloku 3x3 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.17 | | | |
| Tytuł | CanPlaceDigitIndex-  OutOfBoundExceptionTest | **Oczekiwany rezultat** | Rzucenie IndexOutOfBoundException |
| Opis | Sprawdzenie czy funkcja rzuci wyjątek IndexOutOfBoundException | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.18 | | | |
| Tytuł | CanPlaceDigitTestFalse | **Oczekiwany rezultat** | Dla zadanych wartości funkcja CanPlaceDigit zwraca false |
| Opis | Sprawdzenie czy dana liczba występuje w danej kolumnie, wierszu i bloku 3x3 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.19 | | | |
| Tytuł | getUnassignedLocationTrue | **Oczekiwany rezultat** | Para liczb zwracana przez funkcję oraz podana przez testera jest taka sama |
| Opis | Sprawdzenie czy dla danych macierzy(testowane na trzech macierzach) funkcja zwróci dobrą pierwszą lokalizację, do której nie została przypisana liczba | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.20 | | | |
| Tytuł | getUnassignedLocationFalse | **Oczekiwany rezultat** | Para liczb zwracana przez funkcję oraz podana przez testera(inna wolna lokalizacja, ale nie pierwsza) jest inna |
| Opis | Sprawdzenie czy dla danych macierzy(testowane na trzech macierzach) funkcja zwróci dobrą pierwszą lokalizację, do której nie została przypisana liczba | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 1.21 | | | |
| Tytuł | gridToStringTrue | **Oczekiwany rezultat** | Funkcja gridToString zwraca macierz w postaci stringa, według przyjętej przez programistę konwencji |
| Opis | Sprawdzenie czy funkcja poprawnie przekształca macierz na ciąg znaków | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

1. BrutalSolverTest

package pl.sudokusolver.solver;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 2.1 | | | |
| Tytuł | solveSudokuTrueAndEqualsTest | **Oczekiwany rezultat** | Zwrócenie macierzy (rozwiązanych sudoku) identycznych jak dane przez testera, oraz zwrócenie true |
| Opis | Sprawdzenie czy algorytm działa zgodnie z oczekiwaniem, czyli rozwiązuje sudoku według algorytmu brutalnego oraz czy zwraca true, jeżeli udało się rozwiązać sudoku (test na dwóch sudoku, jedno same zera) | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 2.2 | | | |
| Tytuł | solveSudokuTrueAndNotEqualsTest | **Oczekiwany rezultat** | Zwrócenie innych macierzy niż danych przez testera, zwrócenie false bo sudoku udało się rozwiązać |
| Opis | Sprawdzenie czy algorytm działa zgodnie z oczekiwaniem, czyli rozwiązuje sudoku w inny sposób niż tester oraz zwrócenie true, jeżeli sudoku jest możliwe do rozwiązania | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 2.3 | | | |
| Tytuł | solveSudokuFalseTest | **Oczekiwany rezultat** | Funkcja solve zwraca false, ponieważ dostarczone przez testera macierze są niemożliwe do rozwiązania |
| Opis | Sprawdzenie czy funkcja zwraca false dla sudoku, którego nie da się rozwiązać(sprawdzone na 4 sudoku, 2 skrajne przypadki) | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

1. SmartSolverTest

package pl.sudokusolver.solver;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 3.1 | | | |
| Tytuł | solveSudokuTest | **Oczekiwany rezultat** | Zwrócenie macierzy (rozwiązanych sudoku) identycznych jak dane przez testera, oraz zwrócenie true |
| Opis | Sprawdzenie czy algorytm rozwiązuje sudoku według algorytmu SmartSolver | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 3.2 | | | |
| Tytuł | solveSudokuTrueAndEqualsTest | **Oczekiwany rezultat** | Zwrócenie takiej samej macierzy, jak dana przez testera |
| Opis | Sprawdzenie czy algorytm działa zgodnie z oczekiwaniem, dla zerowej macierzy | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 3.3 | | | |
| Tytuł | solveSudokuTrueAndNotEquals | **Oczekiwany rezultat** | Funkcja solve powinna zwrocic solve, ale sudoku powinno zostać rozwiązane w inny sposób niż dane przez testera |
| Opis | Sprawdzenie algorytmu | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 3.3 | | | |
| Tytuł | solveSudokuFalse | **Oczekiwany rezultat** | Funkcja solve zwraca false, ponieważ dostarczone przez testera macierze są niemożliwe do rozwiązania |
| Opis | Sprawdzenie czy funkcja zwraca false dla sudoku, którego nie da się rozwiązać – dwa skrajne przypadki | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

1. SmartSolvePerformanceTest

package pl.sudokusolver.solver;

@ignore

|  |  |  |  |
| --- | --- | --- | --- |
| ID 4.1 | | | |
| Tytuł | SmartSolvePerformanceTest | **Opis i oczekiwany rezultat** | Klasa stworzona po to, żeby przetestować czas rozwiązywania Sudoku algorytmem SmartSolve dla dużej liczby danych. Dostarczamy 159 sudoku do rozwiązania i średni czas dla jednego sudoku wynosi  31.232704402515722 ms  co jest dla nas czasem w pełni zadowalającym, ponieważ w wymaganiach projektu mamy, że czas rozwiązywania dla jednego sudoku powinien wynosić poniżej 7 sekund |

1. CenterLinesComparatorTest

package pl.sudokusolver.recognizerlib.utility.comparators;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 5.1 | | | |
| Tytuł | compareTest | **Oczekiwany rezultat** | Funkcja powinna zwrócić jeden jeśli punkt1.x > punkt2.y, zero jeśli punkt1.x==punkt2.y, minus jeden jeśli punkt1.x<punkt2.y |
| Opis | Sprawdzenie działania funkcji compare z klasy CenterLinesComparator, sprawdzam kilka razy wszystkie trzy możliwe wartości, które może zwracać compare | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

1. PointComparatorTest

package pl.sudokusolver.recognizerlib.utility.comparators;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 6.1 | | | |
| Tytuł | compareTestOnlyY | **Oczekiwany rezultat** | Funkcja powinna zwrócić jeden jeśli (punkt1.y - punkt2.y) >=10, zero jeśli|punkt1.y -punkt2.y|<10,  minus jeden jeśli punkt1.y -punkt2.y <= -10 |
| Opis | Sprawdzenie działania funkcji compare z klasy PointComparator, w tym przypadku testowym porównujemy punkty(x,y), które różnią się tylko wartościami y | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 6.2 | | | |
| Tytuł | compareTestOnlyX | **Oczekiwany rezultat** | Funkcja powinna zwrócić jeden jeśli punkt1.x > punkt2.x, zero jeśli punkt1.x==punkt2.x, minus jeden jeśli punkt1.x<punkt2.x |
| Opis | Sprawdzenie działania funkcji compare z klasy PointComparator, w tym przypadku testowym porównujemy punkty(x,y), które różnią się tylko wartościami x, a różnice między wartościami y obu punktów są mniejsze niż 10 | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 6.3 | | | |
| Tytuł | compareTest | **Oczekiwany rezultat** | Dla zadanych przez testera punktów, funkcja powinna zwracać najpierw trzy 0, potem trzy -1, potem trzy 1 |
| Opis | Sprawdzenie działania funkcji compare z klasy PointComparator, porównujemy punkty testując wszystkie trzy wartości, które może zwracać funkcja compare, funkcja compare porównuje najpierw po y według warunków, z poprzednich przypadków testowych w tej klasie, potem po x | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

1. ImageProcessingTest

package pl.sudokusolver.recognizerlib.utility.staticmethods;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 7.1 | | | |
| Tytuł | procSimpleTest | **Oczekiwany rezultat** | Macierz o jednym wierszu, size\*size kolumnach oraz wszystkich komórkach o wartości 1.0 |
| Opis | Sprawdzenie czy funkcja procSimple działa prawidłowo, czyli czy zwraca macierz o jednym wierszu oraz size\*size kolumnach oraz sprawdzenie w pętli czy zwraca oczekiwane przez nas wartości dla poszczególnych komórek | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz musi być typu CV\_32FC1  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 7.2 | | | |
| Tytuł | centerTest | **Oczekiwany rezultat** | Szerokość i wyskość macierzy po działaniu funkcji, powinna wynosić size – czyli parametr funkcji |
| Opis | Sprawdzenie czy funkcja center przekształca zdjęcie w zdjęcie o określonym rozmiarze równym size\*size | **Poprawność wykonania** | Ok |
| Warunek wstępny | -macierz musi być typu CV\_32FC1  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 7.3 | | | |
| Tytuł | applyMaskTest | **Oczekiwany rezultat** |  |
| Opis |  | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz musi być typu CV\_8UC1  -załadowanie OpenCV | **Kroki** | --- |

1. UtilityTest

package pl.sudokusolver.recognizerlib.utility.staticmethods;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.1 | | | |
| Tytuł | orderPointTest1 | **Oczekiwany rezultat** | Posortowana macierz punktow (x,y) po sumie x+y |
| Opis | Sprawdzenie czy funkcja orderPoints prawidłowo sortuje punkty według sumy x+y i zwraca poprawną macierz | **Poprawność wykonania** | Ok |
| Warunek wstępny | -tylko liczby naturalne  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.2 | | | |
| Tytuł | orderPointTest2 | **Oczekiwany rezultat** | Posortowana macierz punktow (x,y) po sumie x+y |
| Opis | Sprawdzenie czy funkcja orderPoints prawidłowo sortuje punkty według sumy x+y i zwraca poprawną macierz dla innego zestawu danych | **Poprawność wykonania** | Ok |
| Warunek wstępny | -tylko liczby naturalne  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.3 | | | |
| Tytuł | distanceTest1 | **Oczekiwany rezultat** | Funkcja zwraca całkowitą część euklidesowej odleglosci w wektorze dla zadanych punktow |
| Opis | Sprawdzenie czy funkcja distance prawidlowo oblicza odległość między punktami | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |
| ID 8.4 | | | |
| Tytuł | distanceTest2 | **Oczekiwany rezultat** | Funkcja zwraca całkowitą częśc euklidesowej odległości w wektorze dla innych zadanych punktów |
| Opis | Sprawdzenie czy funkcja distance prawidlowo oblicza odległość między punktami | **Poprawność wykonania** | Ok |
| Warunek wstępny | Brak | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.5 | | | |
| Tytuł | ApplyFiltersTest | **Oczekiwany rezultat** | Macierz po wykonaniu funkcji powinna mieć postać jak po nałożeniu filtru NOT |
| Opis | Sprawdzenie poprawności funkcji nakładającej filtry na macierz (sprawdzone filtry NOT oraz brak filtra null) | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.6 | | | |
| Tytuł | applyResizeAndToGrayFiltersTest | **Oczekiwany rezultat** | Macierz po wykonaniu funkcji applyFilters powinna mieć postać jak po wykonaniu filtrów Resize oraz ToGrayFilter dla danych zadanch przez testera |
| Opis | Sprawdzenie poprawności funkcji nakładającej filtry na macierz (sprawdzone filtry Resize i ToGrayFilter) | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID 8.7 | | | | |
| Tytuł | applyFilterCounterTest | **Oczekiwany rezultat** | Ilość nałożonych filtrów powinna się zgadzać z ilością elementów(którymi są obiekty klas z package filters) listy | |
| Opis | Sprawdzenie czy funkcja applyFilters nakłada dokładnie tyle filtrów ile jest w liście, którą przekazujemy do funkcji | **Poprawność wykonania** | Ok | |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- | |
| ID 8.8 | | | | |
| Tytuł | getFileFromResourcesTest | **Oczekiwany rezultat** | | Trzy testy wykonane w tym przypadku powinny zwrócić true, przy sprawdzeniu czy pliki istnieją |
| Opis | Sprawdzenie czy funkcje getANNDump, getSVMDump, getTessData zwracają pliki, które istnieją | **Poprawność wykonania** | | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 8.9 | | | |
| Tytuł | matToBufferedImageTest | **Oczekiwany rezultat** | Zwrócenie odpowiednich typów dla macierzy jednokanałowej i trzykanałowej |
| Opis | Sprawdzanie czy dla macierzy jednokanałowych oraz trzykanałowych po użyciu na macierzy funkcji matToBufferedImage, obiekt typu BufferedImage zwraca dobry typ | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV  - dopuszczalne macierze zwracające channels 1 lub channels 3 | **Kroki** | --- |

1. BlurFilterTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 9.1 | | | |
| Tytuł | blurChangeParametersTest | **Oczekiwany rezultat** | Macierz po użyciu funkcji apply, powinna mieć takie same wartości w komórkach, jak wpisane przez testera |
| Opis | Sprawdzenie czy funkcja apply klasy blurFilter prawidłowo przekształca macierz zgodnie z <https://docs.opencv.org/4.0.1/d4/d86/group__imgproc__filter.html#gaabe8c836e97159a9193fb0b11ac52cf1> | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz powinna być typu CV\_8UC1 lub innego typu z jednym kanałem  - parametry funkcji muszą spełniać założenia  ( size >= 0 && blockSize >= 2 &&  c >= 0 && size % 2 == 1)  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 9.2 | | | |
| Tytuł | blurDefParametersTest | **Oczekiwany rezultat** | Oczekiwana macierz poiwnna posiadać same zera |
| Opis | Sprawdzenie działania funkcji apply z klasy blurFilter | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz powinna być typu CV\_8UC1 lub innego typu z jednym kanałem  - parametry funkcji muszą spełniać założenia  ( size >= 0 && blockSize >= 2 && c >= 0 && size % 2 == 1)  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 9.3 | | | |
| Tytuł | wrongTypeOfMatrixTest | **Oczekiwany rezultat** | Rzucenie wyjątku CvException() przez funkcję apply |
| Opis | Sprawdzenie jak zareaguje funkcja apply na podanie nieprawidłowej macierzy | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 9.4 | | | |
| Tytuł | wrongCreationTest | **Oczekiwany rezultat** | Konstruktor BlurFilter() powinien dla (złych) danych podanych przez testera rzucić wyjątek IllegalArgumentException |
| Opis | Sprawdzenie czy podczas tworzenia obiektu klasy BlurFilter zostanie rzucony wyjątek, jeśli podamy złe dane czyli size < 0 lub blockSize <2 lub c <0 lub size%2!=1 | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. FixedWidthResizeFilterTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID 10.1 | | | | | |
| Tytuł | applyTest | **Oczekiwany rezultat** | | Macierz po użyciu funkcji apply powinien mieć zmienioną wysokość według wzoru height=width\*(input.size().height/input.size().width); | |
| Opis | Sprawdzenie czy funkcja prawidłowo przekształca macierz, czyli proporcjonalnie zmienia wysokość do zmiany szerokośći | **Poprawność wykonania** | | Ok | |
| Warunek wstępny | - macierz powinna być jednokanałowa | **Kroki** | | --- | |
| ID 10.2 | | | | | |
| Tytuł | applyCvExceptionTest | | **Oczekiwany rezultat** | | Rzucenie wyjątku CvException przez funkcję apply |
| Opis | Sprawdzenie, czy funkcja apply rzuci wyjątek dla zmiany szerokosci na ujemną lub złego typu macierz | | **Poprawność wykonania** | | Ok |
| Warunek wstępny | -załadowanie OpenCV | | **Kroki** | | --- |

1. MaxResizeFilterTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 11.1 | | | |
| Tytuł | applyTest | **Oczekiwany rezultat** | Zwrócenie macierzy o takich rozmiarach, jak te które podał tester (sprawdzanie czy funkcja skaluje zgodnie z oczekiwaniami) |
| Opis | Jest to też test funkcji resizeToMaxFilter z klasy ImageProcessing. Funkcja zmienia rozmiar macierzy, skaluje macierz. Test bada wszystkie 4 ścieżki działania funkcji | **Poprawność wykonania** | Ok |
| Warunek wstępny | - parametry funkcji muszą być >0  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 11.2 | | | |
| Tytuł | invalidCreationTest | **Oczekiwany rezultat** | Rzucenie IllegalArgumentException() podczas tworzenia obiektu klasy MaxResizeFilter dla podanych złych parametrów |
| Opis | Sprawdzenie, czy podczas tworzenia obiektu klasy zostanie rzucony wyjątek, jak podamy złe parametry(muszą być > 0) | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. MedianBlurTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 12.1 | | | |
| Tytuł | blurChangeParametersTest | **Oczekiwany rezultat** | Macierz zwraca wartości wpisane przez testera, więcej można znaleźć w dokumentacji openCv  <https://docs.opencv.org/4.0.1/d4/d86/group__imgproc__filter.html#gaabe8c836e97159a9193fb0b11ac52cf1> |
| Opis | Sprawdzenie działania funkcji apply, do której przekazujemy macierz | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz powinna być typu CV\_8U  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 12.2 | | | |
| Tytuł | blurDefParametersTest | **Oczekiwany rezultat** | Oczekiwany rezultat jak w poprzednim przypadku, czyli trzy wartosci w pierwszym wierszu 255.0 oraz w dwoch kolejnych wierszach po trzy 0.0 |
| Opis | Sprawdzenie działania funkcji apply, do której przekazujemy macierz, tworzymy obiekt klasy MedianBlur za pomocą konstruktora bez parametrow | **Poprawność wykonania** | Ok |
| Warunek wstępny | - oczekiwany typ macierzy CV\_8U  -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 12.3 | | | |
| Tytuł | wrongTypeOfMatrixTest | **Oczekiwany rezultat** | Rzucenie CvException przez konstruktor obiektu klasy MedianBlur |
| Opis | Jeśli podamy zły typ macierzy to powinien zostać rzucony wyjątek CvException | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 12.4 | | | |
| Tytuł | wrongCreationTest | **Oczekiwany rezultat** | Rzucenie wyjątku IllegalArgumentException |
| Opis | Jeśli przekazujemy niewłaściwe parametry do funkcji powinien zostać rzucony wyjątek IllegalArgumentException, w tym teście właśnie przekazujemy argumenty, po których ten wyjątek ma wystąpić | **Poprawność wykonania** | Ok |
| Warunek wstępny | - macierz powinna być typu CV\_8U  -załadowanie OpenCV | **Kroki** | --- |

1. NotFilterTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 13.1 | | | |
| Tytuł | applyTest | **Oczekiwany rezultat** | Pierwszy test – uzyskanie macierzy z wartościami 255, drugi test – uzyskanie macierzy z samymi zerami |
| Opis | NotFilter wykonuje tylko bramkę not na macierzy składających się z samych zer, oczekuje otrzymania macierzy składających się z bitów 255, potem otrzymaną macierz znowu przepuszczam przez filtr żeby uzyskać same zera | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV  - macierz powinna być typu CV\_8U | **Kroki** | --- |

1. ResizeFilterTest

|  |  |  |  |
| --- | --- | --- | --- |
| ID 14.1 | | | |
| Tytuł | applyTest | **Oczekiwany rezultat** | Po wykonaniu funkcji apply rozmiar macierzy powinien być taki, jak wpisaliśmy w parametrach |
| Opis | Sprawdzam czy filtr poprawnie zmienia rozmiar macierzy | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 14.2 | | | |
| Tytuł | invalidCreationsTest | **Oczekiwany rezultat** | Rzucenie wyjątku IllegalArgumentException przy każdej próbie stworzenia obiektu ResizeFilter dla size <0 |
| Opis | Nie można ustawić rozmiaru macierzy na < 0, wtedy powinien zostać rzucony wyjątek IllegalArgumentException | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. ToGrayFilterTest

package pl.sudokusolver.recognizerlib.filters;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 15.1 | | | |
| Tytuł | applyTest | **Oczekiwany rezultat** | Po użyciu apply na macierzach, macierze te powinny zwracać funkcja channels() powinna zwracać 1 |
| Opis | ToGrayFilter tak przekształca macierz, żeby ta była jednokanałowa. Sprawdzamy na macierzach wielokanałowych, czy uda się je zmienić na jednokanałowe. | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 15.2 | | | |
| Tytuł | applyTestIfChannel1 | **Oczekiwany rezultat** | Funkcja channel() przed i po wykonaniu apply powinna zwracać to samo |
| Opis | Test sprawdza czy do funkcji da się podać też macierz jednokanałową, wtedy po wykonaniu apply powinna być ona dalej jednokanałowa | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. SudokuTest

package pl.sudokusolver.recognizerlib.sudoku;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 16.1 | | | |
| Tytuł | sudokuOnCreateHaveEmptyGridTest | **Oczekiwany rezultat** | Konstruktor zwróci puste sudoku o wymiarach 9x9, funkcja empty() zwróci true |
| Opis | Test sprawdza, czy sudoku stworzone przez konstruktor bez żadnych parametrów jest macierzą 9x9 wypełnioną zerami, testujemy też w tym teście funkcję empty() | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 16.2 | | | |
| Tytuł | readFromDatTest | **Oczekiwany rezultat** | Dla dwóch poprawnych ścieżek do dwóch plików funkcja readFromDat dla każdej z nich zwraca poprawne Sudoku, takie jak w pliku dat o danym numerze |
| Opis | Test sprawdza czy funkcja readFromDat prawidłowo odczytuje i tworzy sudoku z pliku .dat (są to pliki tekstowe, do których przepisywaliśmy sudoku ze zdjęcia dla łatwiejszego testowania aplikacji) | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV  -pliki .dat z zapisanymi według schematu sudoku | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 16.3 | | | |
| Tytuł | readFromDatExceptionTest | **Oczekiwany rezultat** | Funkcja readFromDat powinna rzucić IOException w dwóch pierwszych testach i w trzecim NullPointerException |
| Opis | Test sprawdza czy dla niepoprawnej ścieżki zostanie rzucony IOException oraz dla wstawionego null w miejsce ścieżki zostanie rzucony NullPointerException | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 16.4 | | | |
| Tytuł | scoreTest | **Oczekiwany rezultat** | Funkcja score powinna zwracać dla każdego przypadku liczbę double taką samą jak obliczona ręcznie dla danego przypadku przez programistę, dopuszczalny błąd double wynosi 1E-5 |
| Opis | Test sprawdza działanie funkcji score, która porównuje jedno sudoku z drugim oraz zwraca procent ich podobieństwa, przeprowadzam tutaj 4 testy, dla identycznych sudoku, dla całkowicie różnych, oraz dwa testy z wysokim i niskim stopniem podobieństwa | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 16.5 | | | |
| Tytuł | toStringTest | **Oczekiwany rezultat** | Zwrócenie przez funkcję toString takiego samego stringa jak podany przez testera |
| Opis | Funkcja powinna przekształcać sudoku na stringa według założonego schematu | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. SimpleRowDataTest

package pl.sudokusolver.recognizerlib.data;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 17.1 | | | |
| Tytuł | getDataFromImgTest | **Oczekiwany rezultat** | Obiekt klasy SimpleRowData powinien zwrcać:  -DataType.Simple  -size równy podanemu w konstruktorze  - ilość wierszy dla dwoch macierzy ma byc identyczna  - poprawną ilość kolumn, dla obu macierzy |
| Opis | W klasie SimpleRowData kluczowa jest funkcja loadFromSheet, która jest wykorzystywana w jednym z konstruktorów SimpleRowData, sprawdzam typ, rozmiar, liczbe kolumn i labels zwracanych przez obiekt tej klasy, stworzony za pomocą konstruktora z prywatną metodą loadFromSheets | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV  - zdjęcie do załadowania | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 17.2 | | | |
| Tytuł | simpleCreationTest | **Oczekiwany rezultat** | Obiekt klasy SimpleRowData powinien zwrcać:  -DataType.Simple  -size równy podanemu w konstruktorze  - ilość wierszy dla dwoch macierzy ma byc identyczna  - ilośc kolumn dla obu macierzy ma byc identyczna  - identyczne wartosci w obcu macierzach |
| Opis | W tym przypadku tworzę obiekt SimpleRowData w inny sposób, za pomocą podania konstruktora dwóch macierzy i rozmiaru, metoda testowania podobna do testu 17.1 + przed przekazaniem do funkcji nadałem macierzom identyczne wartosci, na końcu sprawdzam, czy nic nie zostało przekształcone | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

1. MnistReaderTest

package pl.sudokusolver.recognizerlib.data;

|  |  |  |  |
| --- | --- | --- | --- |
| ID 18.1 | | | |
| Tytuł | mnistFilesDontExistTest | **Oczekiwany rezultat** | Dla dwóch poprawnych ścieżek do dwóch plików funkcja readFromDat dla każdej z nich zwraca poprawne Sudoku, takie jak w pliku dat o danym numerze |
| Opis | Sprawdzamy czy funkcja read rzuci IOException dla nieprawidłowych ścieżek | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 18.2 | | | |
| Tytuł | versionMismatchedExceptionTEst | **Oczekiwany rezultat** | Funkcja dla podanych przez testera danych plików powinna rzucić wyjątek VersionMismattchException |
| Opis | Przekazywane do funkcji pliki powinny mieć tą samą wersję | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV  -pliki .dat z zapisanymi według schematu sudoku | **Kroki** | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| ID 18.3 | | | |
| Tytuł | readCorrectFilesTest | **Oczekiwany rezultat** | Cztery gettery dla dwóch stworzonych macierzy powinny zwrócić wartości oczekiwane przez testera |
| Opis | W tym teście używać poprawnie funkcji read z klasy MNISTReader, potem w testach sprawdzam liczby kolumn i wierszy stworzonych macierzy | **Poprawność wykonania** | Ok |
| Warunek wstępny | -załadowanie OpenCV | **Kroki** | --- |