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
2. SmartSolverPerformanceTest
3. 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 | **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 | **Kroki** | --- |

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

|  |  |  |  |
| --- | --- | --- | --- |
| ID 9.2 | | | |
| Tytuł | blurDefParametersTest | **Oczekiwany rezultat** | Oczekwiana 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) | **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 | Brak | **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 | Brak | **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 | Brak | | **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 | **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 | Brak | **Kroki** | --- |

1. MedianBlurTest
2. NotFilterTest
3. ResizeFilterTest
4. ToGrayFilterTest