

Puzzles

Programowanie obiektowe - projekt końcowy

KRZYSTOF BEDNAREK

27 czerwca 2017

Spis treści

1	Idea programu	2
2	Analiza obiektowa	2
2.1	Wzorzec projektowy	2
2.2	Omówienie klas	2
2.2.1	Model	2
2.2.2	View	3
2.2.3	Controller	3
2.2.4	Klasy dodatkowe	4
3	Diagram klas	5
3.1	Model	5
3.2	Controller	6
3.3	View	6

Idea programu

Główną ideą projektu było stworzenie prostego silnika dla zagadek na kracie. Miał on umożliwiać łatwe zaporogramowanie nowego typu łamigłówki. Program zawiera również przykładową implementację zagadki Wielokropka¹.

Analiza obiektowa

2.1 Wzorzec projektowy

W programie został zastosowany wzorzec projektowy *MVC* (Model - View - Controller).

2.2 Omówienie klas

2.2.1 Model

Puzzle Główną klasą w tej części jest klasa *Puzzle*. Jest ona definicją pojedynczej zagadki i implementuje najważniejsze funkcje scalające poszczególne elementy pojedynczej łamigłówki. Obejmuje to również zapis i odczyt z pliku. Korzysta z klas dziedziczących z *Point*, *Edge* oraz *Field*.

Element Klasa abstrakcyjna *Element* jest wzorcem dla wszystkich podklas przechowujących faktyczne dane na temat konkretnego przykładu danej łamigłówki.

Point, Edge, Field Są to proste klasy mające zapewnić odpowiednią strukturę podklasy klasy *Element*.

BasePoint, BaseEdge, BaseField Są to klasy klas dziedziczących odpowiednio z *Point*, *Edge* oraz *Field*. Mają być wykorzystywane jako domyślne elementy w łamigłówkach nie wykorzystujących jednego z elementów.

Multidot Klasa dziedzicząca po klasie *Puzzle* i będąca głównym klasą główną dla zagadki Wielokropka. W praktyce oznacza to zastąpienie domyślnych pól typów *Point*, *Edge* oraz *Field* obiektami typów odpowiednio *MultidotPoint*, *MultidotEdge* oraz *MultidotField*.

MultidotPoint Klasa dziedziczy z *Point*.

MultidotEdge Klasa dziedzicząca z *Edge*. Przechowuje dane na temat aktualnego oraz przykładowego rozwiązania. Robi poprzez zapisanie czy dana krawędź jest używana lub została oznaczona jako błędna. Implementuje również sposób zapisu i odczytu przykładowego rozwiązania z ciągu znaków.

MultidotField Klasa dziedzicząca z *Field*. Przechowuje dane na temat tego, ile sąsiednich krawędzi powinno zostać użyte podczas rozwiązywania. Implementuje również zapis i odczyt przechowywanych danych do ciągu znaków.

SubclassesOfPuzzle Pomocnicza klasa statyczna. Ma za zadanie znaleźć wszystkie dostępne klasy dziedziczące z *Puzzle*. Dzięki niej po stworzeniu nowego typu łamigłówki nie trzeba ręcznie odnotowywać pojawienia się nowego typu. Ta klasa zrobi to za nas.

¹<http://www.wydawnictwologi.pl/wielokropka>

PuzzleType Pomocnicza klasa pakująca element klasy *Type*. Jest wykorzystywana, aby zmienić domyślny sposób wyświetlania nazwy klasy z pełnej nazwy zawierającej całą ścieżkę na samą, właściwą nazwę.

PuzzleData Pomocnicza klasa służąca do przekazania najważniejszych informacji na zewnątrz obiektu klasy *Puzzle* w postaci pojedynczego obiektu.

2.2.2 View

PuzzlePanel Podstawowa klasa wyświetlająca daną zagadkę. Dziedziczy z *Panel*. Odpytuje obiekty klas *Point*, *Edge*, *Field* zdefiniowane w klasie *Puzzle* o ich wygląd na poszczególnych pozycjach. Następnie składa wyniki w postaci panelu

SingleElementView Klasa *SingleElementView* służy do wygenerowania podstawowych elementów wyglądu poszczególnych wyświetlanych fragmentów. Korzysta z klas dziedziczących z *ElementView* aby uzyskać elementy wyglądu charakterystyczne dla konkretnego typu.

ElementView Klasa abstrakcyjna, będąca wzorem dla klas generujących wygląd, a konkretnie obrazek, znajdujący się na elementach danego typu dziedziczącego z *Element*.

PointView, EdgeView, FieldView Są to klasy abstrakcyjne, dziedziczące z *ElementView*. Zapewniają lepszą strukturalizującą oraz ustalają rozmiar poszczególnych elementów.

BasePointView, BaseEdgeView, BaseFieldView Klasy te dziedziczą odpowiednio z *PointView*, *EdgeView* oraz *FieldView*. Zawierają domyślne definicje wyglądu dla poszczególnych elementów.

MultidotPointView, MultidotEdgeView, MultidotFieldView Klasy te dziedziczą odpowiednio z *PointView*, *EdgeView* oraz *FieldView*. Zawierają definicje wyglądu dla poszczególnych elementów w zagadce Multidot.

MainForm Główna klasa odpowiadająca za zebranie wszystkich podklas i wyświetlenie jako jedno okno. Dziedziczy z *Form*.

PlayPanel Główna klasa odpowiadająca za obsługę podczas wyboru zagadki do rozwiązania oraz samego rozwiązywania. Dziedziczy z *UserControl*. Korzysta z klas *ChoosePuzzle* oraz *SolvePanel*.

ChoosePuzzle Klasa implementująca okno wyboru zagadki do rozwiązania. Dziedziczy z klasy *Panel*. W momencie dokonania wyboru przez użytkownika zostaje uruchomione odpowiednie wydarzenie w klasie *PlayPanel*.

SolvePanel Klasa implementująca okno do obsługi rozwiązywania danej zagadki. Dziedziczy z klasy *Panel*. Korzysta w tym celu z klasy *PuzzlePanel*.

EditPanel Główna klasa odpowiadająca za obsługę podczas tworzenia i edycji zagadek. Dziedziczy z *UserControl*. Korzysta z klas *NewPuzzleView*, *EditPuzzlePanel* oraz *ExistingPuzzleView*.

NewPuzzleView Okno kreatora nowej zagadki. Dziedziczy z klasy *Panel*. Przymuje informacje na temat nazwy, rozmiaru oraz typu zagadki, a następnie wywołuje odpowiednie zdarzenie z klasy *EditPanel*.

ExistingPuzzleView Okno wyboru zagadki do edycji. Dziedziczy z klasy *Panel*. Po dokonaniu wyboru przez użytkownika woła odpowiednie zdarzenie z klasy *EditPanel*.

EditPuzzlePanel Okno odpowiadające za obsługę edycji danej zagadki. Dziedziczy z klasy *Panel*.

2.2.3 Controller

PuzzleContorller Główna klasa zajmująca się zarządzaniem zagadką, czyli wywoływaniem odpowiedniej funkcji obsługującej naciśnięcie na jeden z elementów oraz oceniającej rozwiązanie zagadki.

MultidotContoller Klasa dziedzicząca z *PuzzleContorller*. Implementuje metody obsługi naciśnięcia jednego z elementów oraz oceniające, czy dana zagadka jest rozwiązana.

2.2.4 Klasy dodatkowe

MouseButton Klasa pakująca nazwę przycisku myszki użytego do kliknięcia na element. Została dodana, aby uniknąć wymuszania na użytkowniku obsługi dużo większej klasy wbudowanej oraz umożliwić większą elastyczność w razie zmiany części odpowiadającej za wygląd całości.

PictureBoxImage Klasa pakująca obrazek użyty do wygenerowania wyglądu pojedynczego elementu. Została stworzona z tych samych powodów co klasa *MouseButton*.

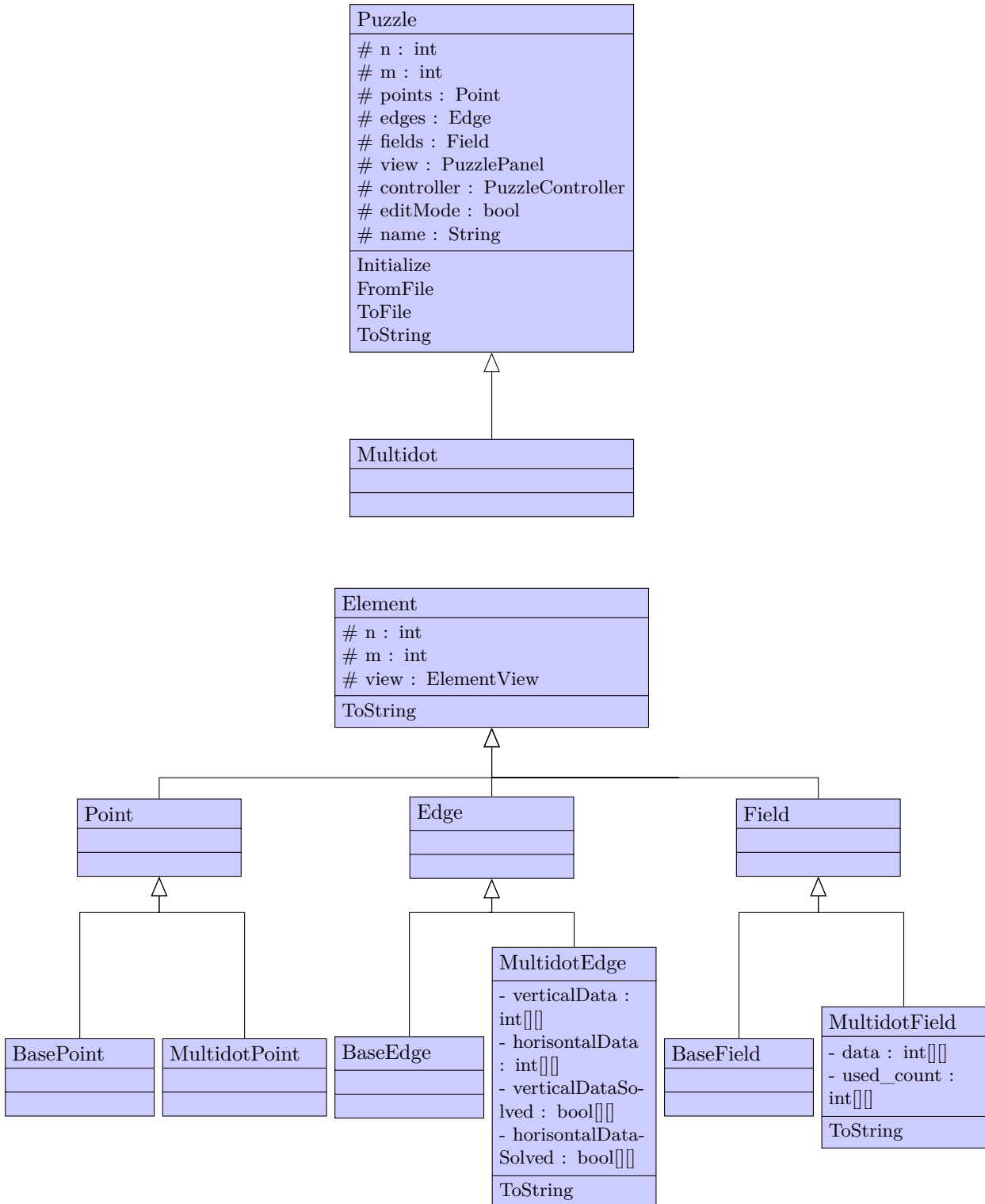
PuzzleArgs Klasa pakująca argumenty przy przekazywane przez klasę *NewPuzzleView* na temat nowej zagadki, która ma być stworzona.

ElementClickArgs Klasa pakująca argumenty przy przekazywane przez klasę *NewPuzzleView* na temat nowej zagadki, która ma być stworzona.

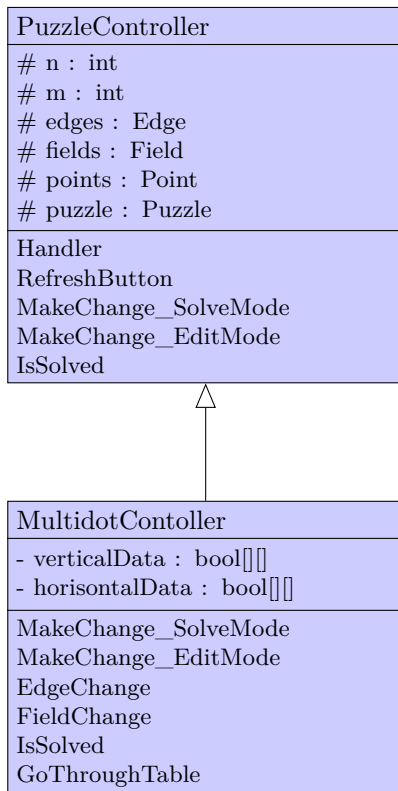
ResizeArgs Klasa pakująca argumenty przy przekazywane przez *MainForm* do *EditPanel* oraz *PlayPanel* na temat zmiany przez nią rozmiaru i wymogu dostosowania się.

Diagram klas

3.1 Model



3.2 Controller



3.3 View

