

# kurs języka Java

## labirynt

Instytut Informatyki  
Uniwersytetu Wrocławskiego

Paweł Rzechonek

---

Zadanie.

Napisz aplikację okienkową w technologii *Swing*, która będzie umożliwiała chodzenie po labiryncie i której celem będzie znalezienie drogi do wyjścia.

Aplikacja ma być formą gry logicznej, wyrabiającej u gracza spostrzegawczość. Zadany jest labirynt w formie prostokątnej siatki, w labiryncie ustawiony jest Podróżnik (początkowo w lewym górnym polu), gracz przesuwa postać Podróżnika z komnaty do komnaty zgodnie z możliwymi przejściami tak, aby dotrzeć do komnaty wyjściowej (w prawym dolnym polu). Rozgrywka może się znajdować w jednym z czterech stanów:

- startowy – przed wygenerowaniem labiryntu,
- w trakcie gry – gracz przesuwa postać z komnaty do komnaty wewnątrz labiryntu,
- koniec z sukcesem – gracz przesunął postać do komnaty wyjściowej,
- koniec z porażką – gracz zrezygnował z odnalezienia drogi do wyjścia bo w jego opinii przejście do wyjścia zostało zamurowane.

Przed rozpoczęciem gry należy wygenerować losowy labirynt oparty na prostokątnej siatce o ustalonych rozmiarach (początkowo  $10 \times 10$ ) i ustawić postać Podróżnika w lewym górnym polu. Układ komnat i przejść między komnatami w labiryncie pamiętaj w odrębnym obiekcie, który będzie pełnił rolę modelu danych dla tego programu. Należy także pamiętać bieżące położenie Podróżnika, ilość wykonanych do tej pory ruchów i czas jaki upłynął od pokazania planszy do momentu wykonania ostatniego przejścia. Punktem wyjścia z labiryntu jest prawe dolne pole. Celem zabawy jest oczywiście doprowadzenie Podróżnika do wyjścia z labiryntu.

Rozpoczęcie gry rozpoczyna się od wygenerowania labiryntu. Wygenerowany labirynt powinien mieć taką właściwość, że pomiędzy dwiema komnatami istnieje dokładnie jedna ścieżka, która je łączy – w języku grafów oznacza to, że labirynt powinien być drzewem.

Dodatkowo po labiryncie krąży Duch, który sukcesywnie psuje labirynt – co sekundę w losowo wybranej komnacie zamurowuje jakieś przejście. Duch pracuje tak długo, dopóki gracz nie zakończy rozgrywki. Działanie Ducha zrealizuj za pomocą obiektu `javax.swing.Timer` – obiekt ten należy uruchomić na początku rozgrywki metodą `start()` a potem po jej zakończeniu zatrzymać metodą `stop()`.

```

javax.swing.Timer timer = new Timer(
    1000,
    new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
            // ...
        }
    });

```

Sama aplikacja powinna składać się z trzech elementów: rozbudowanego menu JMenuBar do sterowania przebiegiem gry (w górnej części okna), skalowalną planszą graficzną JPanel do wizualizacji stanu gry (w części centralnej) i polem tekstowym JTextField zablokowanym do edycji informującym o bieżącym stanie gry (w dolnej części okna). Po każdym wykonanym ruchu należy sprawdzić, czy gra się skończyła. Jeśli tak, to należy rozgrywkę zablokować i wypisać informację o sukcesie; rozgrywka może się również zakończyć poddaniem się gracza – wówczas także należy zablokować rozgrywkę i wypisać informację o porażce (w obu przypadkach zatrzymujemy również Ducha). Oprócz komunikatu o zakończeniu gry pokaż również czas trwania rozgrywki oraz liczbę wykonanych ruchów.

Jako powierzchni kreślarskiej użyj obiektu klasy JPanel. Na panelu tym narysuj labirynt i postać zagubionego Podróżnika (Podróżnik niech będzie zaprezentowany na planszy w postaci obrazka odczytanego z przygotowanego wcześniej pliku graficznego).

```

String filename = "podroznik.jpg";
ImageIcon podróznik = new ImageIcon(filename);
// Image img = podróznik.getImage();

```

Potem w metodzie paintComponent() w klasie dziedziczącej po JPanel użyj metody drawImage() do wyświetlenia tego obrazu na powierzchni kreślarskiej. Oprócz Podróżnika pokaż zamurowane przez Ducha przejścia między komnatami tak, aby się wyróżniały na tle innych ścian.

Ważną częścią aplikacji ma być menu JMenuBar i skróty klawiaturowe. Pasek menu ma się składać z czterech elementów: *Gra*, *Ustawienia*, *Ruchy* i *Pomoc* (ostatni element menu przesunąć do prawej strony):

- W menu *Gra* mają być trzy pozycje (ostatnia pozycja oddzielona separatorem): *Start* (rozpoczęcie nowej gry), *Rezygnacja* (zakończenie gry z porażką) oraz *Koniec* (zamknięcie aplikacji). Akcje związane z tymi zdarzeniami mają mieć przypisane mnemoniki i skróty klawiaturowe odpowiednio: "control S", "control R", "control K".
- W menu *Ustawienia* mają się znaleźć pozycje dotyczące wyglądu i rozmiaru planszy: pozycja z polem wyboru *Oznaczenie pól* (oznacz wiersze kolejnymi literami alfabetu zaczynając od A natomiast kolumny kolejnymi liczbami zaczynając od 1), podmenu *Kolory* do wyboru kolorów a w nim dwie pozycje *Tło* oraz *Ściany* oraz podmenu *Rozmiary* pozwalające określić rozmiaru labiryntu a w nim dwie pozycje *Szerokość* i *Wysokość* (minimalne rozmiary labiryntu to 5x5 a maksymalne to 26x26).

- W menu *Ruchy* mają być cztery pozycje: *W górę* (przesunięcie Podróżnika w górę planszy), *W prawo* (przesunięcie Podróżnika w prawo), *W dół* (przesunięcie Podróżnika w dół planszy), *W lewo* (przesunięcie Podróżnika w lewo). Akcje związane z ruchami Podróżnika mają mieć przypisane ikonki oraz skróty klawiaturowe odpowiednio: `KeyEvent.VK_UP`, `KeyEvent.VK_RIGHT`, `KeyEvent.VK_DOWN`, `KeyEvent.VK_LEFT`.
- Na końcu ostatnie menu *Pomoc* (dosunięte do prawego brzegu okna) z dwiema pozycjami: *O aplikacji* (zasady gry) i *O autorze* (informacje o twórcy programu); wybór każdej z nich powinien powodować pojawienie się dialogowego okienka modalnego z odpowiednimi informacjami.

Zadbaj o estetyczny wygląd aplikacji!