Politechnika Świętokrzyska w Kielcach	
PROJEKT JAVA	
TEMAT: **Gra Tetris**	**Grupa 2ID12B** Aleksandra Minkina Konrad Smoląg

Tematem naszego projektu było stworzenie gry typu Tetris.
Rozpoczyna się ona na prostokątnej planszy, która z początku jest pusta.
W trakcie gry z górnej krawędzi planszy, zaczynają się pojawiać pojedynczo klocki (tetrominos). Spadają one do dolnej krawędzi. Kiedy spadający klocek zostanie umieszony na dole, z góry zaczyna lecieć następny. Naszym zadaniem jest układanie tetrimino na planszy w taki sposób, aby kwadraty składające się na nie utworzyły wiersz na całej szerokości prostokąta. W takiej sytuacji wiersz ten zostaje usunięty, a pozostałe klocki opadają w kierunku dna, tworząc więcej przestrzeni dla następnych elementów. Jeżeli natomiast klocki zostaną ułożone na tyle wysoko, że następny tetromino nie będzie mógł sie pojawić następuje koniec gry zasygnalizowany napisem „ Koniec gry ‘’ u dołu okna.

W naszej grze możemy obracać klockiem, używając strzałek klawiatury.
- Strzałka w górę - obraca tetromino w lewo do okoła osi.
- Strzałka w dół - przyspiesza spadanie o jedną linię
- Spacja - „zrzuca” klocek na dolną krawędź.
Grę można również spauzować klawiszem „P”.

W pakiecie występują 3 klasy: Shape, Board oraz Tetris.
W klasie Shape użyliśmy biblioteki java.util.Random, która tworzy generator liczb losowych przydatnych do losowania klocków spadających z góry.
W Board zastosowaliśmy takie biblioteki jak:
- java.awt.Color, java.awt.Dimension, java.awt.Graphics – kolory, rozmiary, rysowanie.
- java.awt.event.ActionEvent, java.awt.event ActionListner, java.awt.event KeyAdapter, java.awt.event KeyEvent – obsługa zdarzeń, przycisków.
- jawax.swing.JPanel, jawax.swing.JLabel, jawax.swing.Timer

Przesuwanie klocka:

```
private boolean tryMove(Shape newPiece, int newX, int newY) {
                        for (int i = 0; i < 4; ++i) {
                                int x = newX + newPiece.x(i);
                                int y = newY - newPiece.y(i);

                                if(x < 0 || x>= BOARD_WIDTH || y < 0 || y >= BOARD_HEIGHT)
                                        return false;

                                if(shapeAt(x, y) != Tetrominos.NoShape)
                                        return false;



                        }
                        curPiece = newPiece;
                        curX= newX;
                        curY= newY;
                        repaint();

                        return true;

        }
```

„Zrzucenie” klocek na dolną krawędź:

```
private void dropDown (){
                int newY = curY;

                while (newY > 0) {
                        if (!tryMove(curPiece, curX, newY - 1))
                                break;

                        --newY;
                }
                pieceDropped();

                }
```

Klasa odpowaidająca za przyciski:

```
class MyTetrisAdapter extends KeyAdapter {
                @Override
                public void keyPressed(KeyEvent ke){
                        if (!isStarted || curPiece.getShape() == Tetrominos.NoShape) {
                                return;
                        }
                        int keyCode = ke.getKeyCode();

                        if(keyCode == 'p' || keyCode == 'P')
                                pause();

                        if(isPaused)
                                return;

                        switch (keyCode) {
                        case KeyEvent.VK_LEFT:
                                tryMove(curPiece, curX -1, curY);
                                break;
                        case KeyEvent.VK_RIGHT :
                                tryMove(curPiece, curX + 1, curY);
                                break;

                        case KeyEvent.VK_UP:
                                tryMove(curPiece.rotateLeft(), curX, curY);
                                break;
                        case KeyEvent.VK_SPACE:
                                dropDown();
                                break;

                        case KeyEvent.VK_DOWN:
                                oneLineDown();
                                break;



                        }
        }

        }
```

Start:

```
public void start() {
                if (isPaused)
                        return;

                isStarted = true;
                isFallingFinished = false;
                numLinesRemoved = 0;
                clearBoard();
                newPiece();
                timer.start();
        }
```

Pauza:

```
public void pause(){
        if (!isStarted)
                return;

        isPaused =!isPaused;

        if(isPaused) {
                timer.stop();
                statusBar.setText("Pauza");
        } else {
                timer.start();
                statusBar.setText(String.valueOf(numLinesRemoved));


        }

        repaint();
}
```

Obrócenie klocka:

```
public Shape rotate() {
        if (pieceShape == Tetrominos.SquareShape)
                return this;

        Shape result = new Shape();
        result.pieceShape = pieceShape;

        for (int i =0; i<4; i++) {
                result.setX(i , y(i));
                result.setY(i, -x(i));
        }
                return result;

        }
```

Ilość pracy włożona przez poszczególnych członków zespołu jest porównywalna.