

<p style="text-align: center;">Politechnika Świętokrzyska Wydział Elektrotechniki, Automatyki i Informatyki Kierunek Informatyka Studia Stacjonarne I Stopnia Rok 2</p>	
Grupa Dziekańska: 2ID14A	Imię i Nazwisko: Marcin Majdański
Temat Projektu: Gra Saper w Javie	Nazwa Przedmiotu: Programowanie obiektowe (Java) Projekt

1. Ogólny Opis Projektu

Mój projekt to prosta gra „Saper” napisana w języku Java z użyciem biblioteki JavaFX, która pozwala na tworzenie aplikacji graficznych w środowisku Java. Do zrobienia projektu wykorzystałem IDE: IntelliJ oraz bibliotekę JavaFX wersję 13(najnowszą) i wtyczkę(plugin) Maven.

2. Funkcjonalność Projektu

Tematem projektu była gra komputerowa o nazwie „Saper”, projekt po skompilowaniu i odpaleniu wyświetla się na ekranie, jako okno wypełnione kafelkami. Program obsługuje się tylko i wyłącznie myszką, klikając w obrębie okna.

3. Uruchomienie i Obsługa Projektu

Żeby uruchomić projekt należy zaopatrzyć się w IDE IntelliJ dostępne na stronie [jetbrains.com](https://www.jetbrains.com). Następnie w folderze projektu trzeba wybrać pliki o rozszerzeniu *.iml. Kiedy projekt się uruchomi i wszystko się załaduje należy wybrać z górnej list *Run*→*Run...* lub kombinację klawiszy: *Alt+Shift+F10*.

Projekt po uruchomieniu obsługuje się myszką klikając lewym lub prawym przyciskiem myszy na kafelki.

4. Stworzone Klasy, Metody, Funkcje

- 4.1. **public class App extends Application** – Jest to główna klasa całej aplikacji, w której są wszystkie inne klasy. Podpięta jest pod bibliotekę javaFX znajdują się w niej wszystkie klasy, metody i zmienne aplikacji.
- 4.2. **private Parent createContent()** – Funkcja ta tworzy planszę na której jest wyświetlana gra, oraz wypełnia planszę bombami i informacjami o tym ile bomb znajduje się w sąsiedztwie każdego kafelka. Zwraca obiekt root
- 4.3. **private List<Kafelki> znajdzSasiadow(Kafelki kafelek)** – Funkcja ta jest wywoływana przez powyższą i służy do sprawdzenia każdego kafelka pod względem sąsiedztwa bomb. Jako parametr przyjmuje 'kafelek' który jest obiektem wskazującym na konkretny kafelek. Zwraca obiekt 'sąsiedzi' z informacją o bombach w sąsiedztwie kafelka.
- 4.4. **public class Kafelki extends StackPane** – Klasa ta odpowiada za to jak wyświetlane są wszystkie kafelki na ekranie w oknie aplikacji.
- 4.5. **public Kafelki(x,y,Bomba)** – jest to konstruktor powyższej klasy. Parametry, to współrzędne kafelka oraz zmienna logiczna informująca czy w kafelku jest bomba(wtedy przyjmuje wartość *true*) lub czy jej nie ma(wartość *false*). Tutaj jest określone jak mają wyglądać kafelki i jak mają się zachowywać przy kliknięciu na nie
- 4.6. **public void open()** – jest to metoda klasy Kafelki, która definiuje jak się zachowa kafelek przy kliknięciu lewym przyciskiem myszy
- 4.7. **public void zaznacz()** – ta metoda definiuje jak się zachowa kafelek po kliknięciu prawym przyciskiem myszy
- 4.8. **public void start(Stage stage)** – funkcja stworzona automatycznie przez IDE w momencie tworzenia projektu, odpowiada za pokazanie okna aplikacji na ekranie
- 4.9. **public static void main(String[] args)** – funkcja utworzona automatycznie w trakcie tworzenia projektu, odpowiada tylko i wyłącznie za uruchomienie całej aplikacji

Projekt robiłem sam więc wszystko co zostało zrobione, było zrobione przeze mnie.