МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ**

**ЛАБОРАТОРНА РОБОТА №5**

З дисципліни:

«Системний аналіз та проєктування комп’ютерних інформаційних систем»

Виконав студент I (скороченого) курсу

групи КН-333а

Смотрич Артем Жаннович

Одеса, 2022

**Задача:**

Створити функцію, що генерує шахматну дошку. При цьому можна використовувати будь-які теги за власним бажанням. Дошка має бути розлінована відповідним чином, тобто чергувати чорні та білі клітинки. Рядки повинні нумеруватися числами від 1 до 8, стовпці – латинськими літерами A, B, C, D, E, F, G, H. Заповнити створену таблицю шахматними фігурами.

**Хід роботи**

**Аналіз задачі та алгоритм її рішення.**

**Аналіз задачі:**

Найефективнішим шляхом рішення цієї задачі буде використання контейнеру GridView у якому у комірки буде додаватися певна шахова фігура. Додавання буде реалізоване у звичайних циклах. У двох масивах буде зберігатися окремо розташування фігур, у цих масивах цифри це назва файлу зображення.

***Алгоритм рішення:***

1. Підготувати форму з Grid Pane.
2. Сформувати масиви, який містить назву картинки чорних та білих фігур окремо.
3. У циклі пройти всі комірки за умовою, що комірка буде темного кольору випадку коли комірка:
   1. у непарному рядку і парному стовпці.
   2. у парному рядку і непарному стовпці.
4. Додати у комірку об’єкт многокутник.
5. Заповнити крайні праву та ліву сторони цифрами від одного до восьми.
6. Заповнити крайні верхню і нижню сторони буквами від A до H.
7. Заповнити через цикл за допомогою створених масивів з іменами картинок чорних та білих фігур.

**Обґрунтування мови програмування для вирішення задачі.**

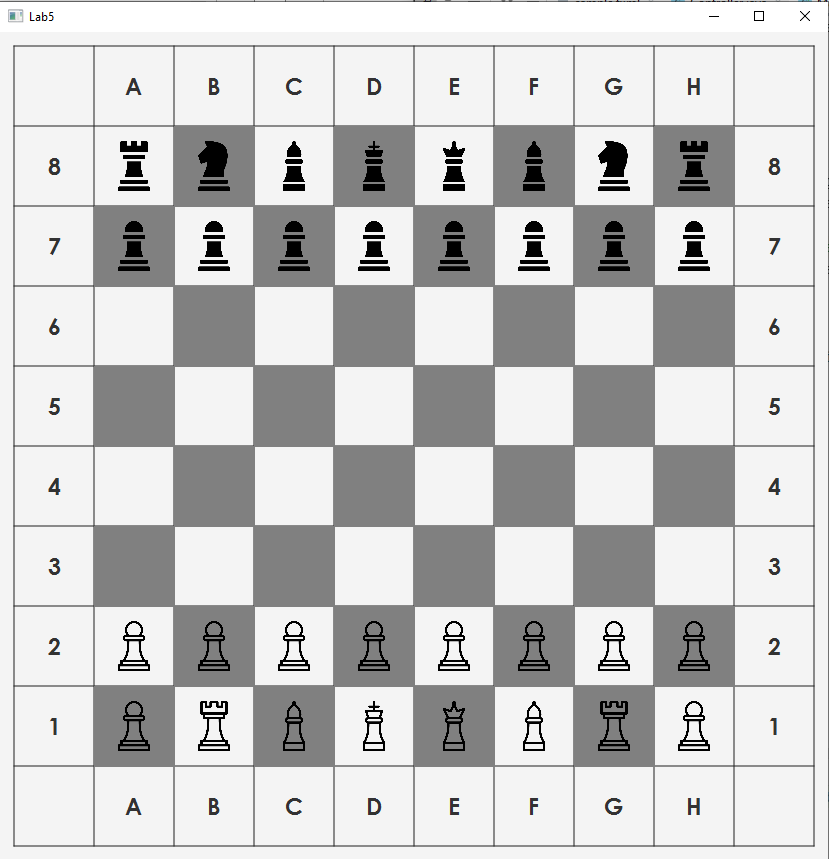
Вирішення поставленої задачі не потребує певної мови програмування, тому мій вибір впав на Java, як найбільш відому для мене мову, яка, крім цього, є одною з найпопулярніших мов програмування. Для створення GUI був використаний JavaFx, тому що він найзручніший з існуючих графічних бібліотек Java.

**Результати**

***Лістинг програми:***

public class Controller implements Initializable{  
 public GridPane grid;  
 @Override  
 public void initialize(URL url, ResourceBundle resourceBundle) {  
 for(int i = 1; i < 9; i++){  
 for (int j = 1; j < 9; j++){  
 if(i%2!=0){  
 if(j%2==0){  
 grid.add(rectangle(), j, i);  
 }  
 }else{  
 if(j%2!=0){  
 grid.add(rectangle(), j, i);  
 }  
 }  
  
 }  
 }  
 char sign = 'A';  
 int num = 8;  
 for(int i = 1; i < 9; i++){  
 grid.add(label(String.*valueOf*(sign)), i,0 );  
 grid.add(label(String.*valueOf*(sign)), i,9 );  
  
 grid.add(label(String.*valueOf*(num)), 9, i );  
 grid.add(label(String.*valueOf*(num)), 0, i );  
 sign++;  
 num--;  
 }  
 int[] white = {2, 4, 8, 10, 12, 8, 4, 2};  
 int[] black = {3, 5, 7, 9, 11, 7, 5, 3};  
   
 for (int i = 0; i < 8; i++){  
 grid.add(image(String.*valueOf*(black[i])), i+1, 1);  
 grid.add(image(String.*valueOf*(white[i])), i+1, 8);  
  
 grid.add(image("1"), i+1, 2);  
 grid.add(image("2"), i+1, 7);  
 }  
 }  
 public ImageView image(String s){  
 Image image = new Image("file:"+s+".png");  
 ImageView imageView = new ImageView(image);  
 imageView.setFitHeight(50);  
 imageView.setFitWidth(50);  
 GridPane.*setHalignment*(imageView, HPos.*CENTER*);  
 return imageView;  
 }  
 public Rectangle rectangle(){  
 Rectangle rect = new Rectangle(80, 80);  
 rect.setFill(Color.*GRAY*);  
 rect.setStroke(Color.*GRAY*);  
 return rect;  
 }  
 public Label label(String str){  
 Label label = new Label(str);  
 label.prefWidth(50);  
 label.setFont(Font.*font*("Century Gothic", FontWeight.*BOLD*, 23));  
 GridPane.*setHalignment*(label, HPos.*CENTER*);  
 return label;  
 }  
}

**Результат виконання програми:**

**