```
 private void izracunaj() {
        Task task = getCalculateTask();
        //ponovo učitaj progres bar kad se ponovo učita progres task-a
        progressBar.progressProperty().bind(task.progressProperty());
        //task koji je u novoj dretvi
        new Thread(task).start();
        // kada se novi task zavrsi onda postavi tekst na rezultat
        task.setOnSucceeded(event -> {
            Object result = task.getValue();
            if (result == null) {
                textField.setText("Error");
                Timer timer = new Timer(3000, e1 -> {
                    if (textField.getText().equals("Error"))
                        textField.setText("");
                });
                timer.setRepeats(false);
                timer.start();
            }
            else textField.setText(result.toString());
        });
    }


    private Task getCalculateTask() {
        Task task = new Task<Number>() {
            @Override
            public Number call() {
                int max = 5;
                // namjerno čekamo pola sekunde da se prikaze progres
                for (int i = 1; i < max; i++) {
                    try {
                        Thread.sleep(500);
                    } catch (InterruptedException e) {
                    }
                    updateProgress(i, max);
                }
                Number result = null;
```

```java
                try {
                    result = calculator.getResultFromString(textField.getText());
                } catch (Exception e) {
                }
                updateProgress(max, max);
                return result;
            }
        };
        return task;
    }


}

```