

## 10. Deveta laboratorijska vježba

### 10.1. Tema vježbe

Svrha laboratorijske vježbe je korištenje višenitnosti u JavaFX aplikaciji tako da se paralelno s korištenjem aplikacije mogu odvijati i druge radnje. Aplikacija mora implementirati funkcionalnosti prikazane na sljedećem YouTube videu: <https://www.youtube.com/watch?v=ESuXl1Pm8Ws>.

### 10.2. Zadatak za pripremu

Proširiti rješenje devete laboratorijske vježbe na način kako je opisano u sljedećim koracima:

1. Kopirati projekt iz devete laboratorijske vježbe i preimenovati ga u na način da mu se promijeni redni broj u „10“.
2. Kreirati paket pod nazivom „**hr.java.vjezbe.niti**“.
3. Unutar paketa „**hr.java.vjezbe.niti**“ kreirati klasu „**DatumRodjenjaNit**“ koja će omogućavati dohvaćanje listu studenata koji na trenutni datum imaju rođendan i prikazivati dijalog o tome svakih deset sekundi prilikom rada aplikacije. To je moguće implementirati korištenjem sljedećeg programskog isječka kod pokretanja aplikacije:

```
Timeline prikazSlavljenika = new Timeline(  
new KeyFrame(Duration.seconds(10), new EventHandler<ActionEvent>() {  
    @Override  
    public void handle(ActionEvent event) {  
        Platform.runLater(new DatumRodjenjaNit());  
    }  
}));  
prikazSlavljenika.setCycleCount(Timeline.INDEFINITE);  
prikazSlavljenika.play();
```

Lista studenata koja na trenutni datum imaju rođendan može se dohvatiti izvršavanjem sljedećeg upita:

```
SELECT * FROM STUDENT WHERE DAY(DATUM_RODJENJA) = DAY(?) AND MONTH(DATUM_RODJENJA) =  
MONTH(?)
```

4. Unutar paketa „**hr.java.vjezbe.niti**“ kreirati klasu „**NajboljiStudentNit**“ koja korištenjem metoda „**BazaPodataka.dohvatiStudentePremaKriterijima**“ i „**BazaPodataka.dohvatiIspitePremaKriterijima**“ omogućavati određivanje studenta s najboljim prosjekom na vrhu ekrana. Nakon što se dohvate ispiti za pojedinog studenta, prosjek ispita se može određivati ovim izrazom:

```
OptionalDouble prosjek = listaIspitaZaStudenta.stream().mapToDouble(i ->  
i.getOcjena()).average();
```

Nit „**NajboljiStudentNit**“ potrebno je pokrenuti nakon pokretanja aplikacije i izvršavati svake tri sekunde na sličan način kao i nit „**DatumRodjenjaNit**“.

## NAPOMENE:

- 1) Osim implementacija vježbe prema uputama, dozvoljeno je uvoditi i promjene ako su opravdane i ne narušavaju koncepte objektno-orijentiranog programiranja.
- 2) S obzirom da je potrebno na dva mjesta kreirati objekte klase „Student“ na temelju podataka iz baze (unutar metode „BazaPodataka.dohvatiStudenteKojimajuRodjendan“ i „BazaPodataka.dohvatiStudentePremaKriterijima“) preporuča se logiku „mapiranja“ podataka iz baze na varijable objekta izdvojiti u zasebnu metodu i tu metodu pozivati na ta dva mjesta:

```
private static Student mapirajRezultatNaStudenta(ResultSet rezultati) throws SQLException {  
    Long id = rezultati.getLong("id");  
    String jmbag = rezultati.getString("jmbag");  
    String ime = rezultati.getString("ime");  
    String prezime = rezultati.getString("prezime");  
    LocalDate datumRodjenja =  
        rezultati.getTimestamp("datum_rodjenja").toInstant()  
            .atZone(ZoneId.systemDefault()).toLocalDate();  
    Student noviStudent = new Student(id, ime, prezime, datumRodjenja, jmbag);  
    return noviStudent;  
}
```