Programiranje mobilnih uređaja

Implementacija sustava za upravljanje zoološkim vrtom

Petar Banjak 3.G

Elektrotehnička i ekonomska škola

Nova Gradiška

Sadržaj

[**Uvod** 2](#_Toc157548288)

[**Dizajn glavnog sučelja** 3](#_Toc157548289)

[**Programski kod glavnog sučelja** 4](#_Toc157548290)

[**Programski kod „item\_employee“** 10](#_Toc157548291)

[**MainActivity** 17](#_Toc157548292)

[**Data class „employees“** 20](#_Toc157548293)

[**Class „employees\_adapter“** 21](#_Toc157548294)

# **Uvod**

U Android Studiu potrebno je bilo implementirati sustav za upravljanje zoološkim vrtom s nastambama, životinjama i zaposlenicima zaduženim za pojedinu nastambu upotrebom klasa, firebase-a i RecyclerView-a

# **Dizajn glavnog sučelja**

****

***Dizajn glavnog sučelja***

Dizajn se sastoji od 4 editTexta, buttona i RecyclerView-a

U 4 editTexta unosimo podatke: ime, prezime, godine i radni položaj

Pomoću buttona unosimo podatke u RecyclerView

Preko RecyclerView-a gledamo unesene podatke

## **Programski kod glavnog sučelja**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod button-a „Unesi“***

**id –** označava id našeg gumba, u ovom slučaju to je „commit“

**layout\_width –** određuje širinu našeg gumba; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u gumbu

**layout\_height -** određuje visinu našeg gumba; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u gumbu

**layout\_constraintBottom\_toTopOf –** određuje nam s čime je poravnat donji dio našeg gumba; u našem slučaju s gornjim dijelom RV-a „recycler“

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg gumba; u ovom slučaju s desnim dijelom zaslona

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg gumba; u ovom slučaju s lijevom stranom ekrana „parent“

**layout\_constraintTop\_toBottomOf -** određuje nam s čime je poravnat gornji dio našeg gumba; u našem slučaju s donjim dijelom editText „radniPolozaj“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod editText „ime“***

**id –** označava id našeg editText-a, u ovom slučaju to je „ime“

**layout\_width –** određuje širinu našeg editText-a; postavljen je na 0dp što znači da će editText zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg editText-a; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u editText

**hint –** iz resursa string.xml uzima tekst koji smo dodijelili stringu brzina, u našem slučaju to je tekst „Unesi ime“ i on će biti ispisan kada ništa nije upisano u EditText-u

**textColorHint –** boja teksta za hint

**marginTop –** gornja margina

**textAllignment –** poravnanje teksta; postavljeno na sredinu

**textSize –** veličina teksta

**layout\_constraintEnd\_toStartOf –** određuje nam s čime je poravnat kraj (desni dio) našeg editText; u ovom slučaju s početkom (lijevim dijelom) gumba „commit“

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg editTexta; u ovom slučaju s lijevom stranom recycler-a „recycler“

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg editText-a; u našem slučaju s gornjim dijelom ekrana „parent“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod editText „prezime“***

**id –** označava id našeg editText-a, u ovom slučaju to je „prezime“

**layout\_width –** određuje širinu našeg editText-a; postavljen je na 0dp što znači da će editText zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg editText-a; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u editText

**hint –** iz resursa string.xml uzima tekst koji smo dodijelili stringu brzina, u našem slučaju to je tekst „Unesi prezime“ i on će biti ispisan kada ništa nije upisano u EditText-u

**textColorHint –** boja teksta za hint

**textSize –** veličina teksta

**layout\_constraintBottom\_toTopOf -** određuje nam s čime je poravnat donji dio našeg editText-a; u našem slučaju s gornjim dijelom recycler-a „recycler“

**layout\_constraintEnd\_toStartOf –** određuje nam s čime je poravnat kraj (desni dio) našeg editText; u ovom slučaju s početkom (lijevim dijelom) gumba „commit“

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg editTexta; u ovom slučaju s lijevom stranom editText-a „ime“

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg editText-a; u našem slučaju s gornjim dijelom editText-a „ime“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod editText „godine“***

**id –** označava id našeg editText-a, u ovom slučaju to je „godine“

**layout\_width –** određuje širinu našeg editText-a; postavljen je na 0dp što znači da će editText zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg editText-a; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u editText

**hint –** iz resursa string.xml uzima tekst koji smo dodijelili stringu brzina, u našem slučaju to je tekst „Unesi godine“ i on će biti ispisan kada ništa nije upisano u EditText-u

**textColorHint –** boja teksta za hint

**textSize –** veličina teksta

**layout\_marginStart ­–** početna margina (lijeva)

**layout\_constraintBottom\_toBottomOf -** određuje nam s čime je poravnat donji dio našeg editText-a; u našem slučaju s donjim dijelom editText-a „ime“

**layout\_constraintEnd\_toStartOf –** određuje nam s čime je poravnat kraj (desni dio) našeg editText; u ovom slučaju s početkom (lijevim dijelom) recycler-a „recycler“

**layout\_constraintStart\_toEndOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg editTexta; u ovom slučaju s lijevom stranom editText-a „ime“

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg editText-a; u našem slučaju s gornjim dijelom editText-a „ime“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod editText „radniPolozaj“***

**id –** označava id našeg editText-a, u ovom slučaju to je „radniPolozaj“

**layout\_width –** određuje širinu našeg editText-a; postavljen je na 0dp što znači da će editText zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg editText-a; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u editText

**hint –** iz resursa string.xml uzima tekst koji smo dodijelili stringu brzina, u našem slučaju to je tekst „Unesi godine“ i on će biti ispisan kada ništa nije upisano u EditText-u

**textColorHint –** boja teksta za hint

**textSize –** veličina teksta

**layout\_marginStart ­–** početna margina (lijeva)

**layout\_constraintBottom\_toBottomOf -** određuje nam s čime je poravnat donji dio našeg editText-a; u našem slučaju s donjim dijelom editText-a „prezime“

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg editText; u ovom slučaju s krajem (desnim dijelom) editText-a „godine“

**layout\_constraintStart\_toEndOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg editTexta; u ovom slučaju s desnom stranom editText-a „prezime“

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg editText-a; u našem slučaju s gornjim dijelom editText-a „prezime“

Slika na kojoj se prikazuje tekst, snimka zaslona, izbornik, Font

Opis je automatski generiran

***Programski kod recyclerView-a „recycler“***

**id –** označava id našeg recycler-a, u ovom slučaju to je „recycler“

**layout\_width –** određuje širinu našeg recycler-a; postavljen je na 0dp što znači da će gumb zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg recycler-a; postavljen je na wrap\_content što znači da će gumb zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u recycler

**layout\_marginStart ­–** početna margina (lijeva)

**layout\_marginTop ­–** gornja margina

**layout\_marginEnd ­–** krajnja margina (desna)

**layout\_marginBottom ­–** donja margina

**layout\_constraintBottom\_toBottomOf -** određuje nam s čime je poravnat donji dio našeg recycler-a; u našem slučaju s donjim dijelom ekrana

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg recycler-a; u ovom slučaju s desnim dijelom ekrana

**layout\_constraintStart\_toEndOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg recycler-a; u ovom slučaju s lijevom stranom ekrana

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg recycler-a; u našem slučaju s donjim dijelom editText-a „ime“

# **Programski kod „item\_employee“**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod textView-a „id“***

**id –** označava id našeg editText-a, u ovom slučaju to je „id“

**layout\_width –** određuje širinu našeg textView; postavljen je na wrap\_content što znači da će textView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u textView-a

**layout\_height -** određuje visinu našeg textView -a; postavljen je na wrap\_content što znači da će textView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst u textView

**textSize –** veličina teksta

**textColor –** boja teksta

**layout\_weight –** težina elementa

**gravity -** poravnanje

**layout\_margin ­–** sve 4 margine

**layout\_constraintBottom\_toTopOf -** određuje nam s čime je poravnat donji dio našeg textView -a; u našem slučaju s donjim dijelom linearLayout „ll“

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg textView; u ovom slučaju s krajem (desnim dijelom) linearLayout „ll2“

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg textView-a; u ovom slučaju s desnom stranom linearLayout „ll2“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod Linear layout-a „ll“***

**id –** označava id našeg Linear layout-a, u ovom slučaju to je „ll“

**layout\_width –** određuje širinu našeg Linear layout-a; postavljen je na match\_parent što znači da će Linear layout zauzeti mjesto preko cijele širine ekrana

**layout\_height -** određuje visinu našeg Linear layout-a; postavljen je na wrap\_content što znači da će Linear layout zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_margin ­–** sve 4 margine

**orientation –** orijentacija Linear layout-a

**layout\_constraintBottom\_toBottomOf -** određuje nam s čime je poravnat donji dio našeg Linear layout-a; u našem slučaju s donjim dijelom linearLayout „ll2“

**layout\_constraintTop\_toTopOf –** određuje nam s čime je poravnat gornji dio našeg Linear layout-a; u ovom slučaju s gornjom stranom zaslona

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Multimedijski softver

Opis je automatski generiran

***Programski kod TextView-a „ime“***

**id –** označava id našeg TextView -a, u ovom slučaju to je „ime“

**layout\_width –** određuje širinu našeg TextView -a; postavljen je na 0dp što znači da će TextView zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg TextView -a; postavljen je na wrap\_content što znači da će TextView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_weight –** težina elementa

**layout\_margin –** veličina sve 4 margine

**textColor –** boja teksta

**text –** tekst

**gravity -** poravnanje

**textSize –** veličina teksta

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg TextView; u ovom slučaju s desnom stranom ekrana

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg TextView-a; u ovom slučaju s lijevom stranom ekrana

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg TextView -a; u našem slučaju s gornjim dijelom ekrana „parent“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod TextView-a „prezime“***

**id –** označava id našeg TextView -a, u ovom slučaju to je „prezime“

**layout\_width –** određuje širinu našeg TextView -a; postavljen je na 0dp što znači da će TextView zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg TextView -a; postavljen je na wrap\_content što znači da će TextView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_weight –** težina elementa

**layout\_margin –** veličina sve 4 margine

**textColor –** boja teksta

**text –** tekst

**gravity -** poravnanje

**textSize –** veličina teksta

**layout\_constraintEnd\_toEndOf –** određuje nam s čime je poravnat kraj (desni dio) našeg TextView; u ovom slučaju s desnom stranom ekrana

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat početak (lijeva strana) našeg TextView-a; u ovom slučaju s lijevom stranom ekrana

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg TextView -a; u našem slučaju s gornjim dijelom TextView-a „ime“

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod Linear layout-a „ll2“***

**id –** označava id našeg Linear layout-a, u ovom slučaju to je „ll2“

**layout\_width –** određuje širinu našeg Linear layout-a; postavljen je na match\_parent što znači da će Linear layout zauzeti mjesto preko cijele širine ekrana

**layout\_height -** određuje visinu našeg Linear layout-a; postavljen je na wrap\_content što znači da će Linear layout zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_margin ­–** sve 4 margine

**orientation –** orijentacija Linear layout-a

**layout\_constraintBottom\_toBottomOf -** određuje nam s čime je poravnat donji dio našeg Linear layout-a; u našem slučaju s donjim dijelom ekrana

**layout\_constraintStart\_toStartOf –** određuje nam s čime je poravnat lijevi dio našeg Linear layout-a; u ovom slučaju s lijevom stranom Linear layout-a „ll“

**layout\_constraintTop\_toTopOf -** određuje nam s čime je poravnat gornji dio našeg Linear layout-a; u našem slučaju s gornjim dijelom ekrana

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran***Programski kod TextView-a „godine“***

**id –** označava id našeg TextView -a, u ovom slučaju to je „godine“

**layout\_width –** određuje širinu našeg TextView -a; postavljen je na 0dp što znači da će TextView zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg TextView -a; postavljen je na wrap\_content što znači da će TextView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_weight –** težina elementa

**layout\_margin –** veličina sve 4 margine

**textColor –** boja teksta

**text –** tekst

**gravity -** poravnanje

**textSize –** veličina teksta

Slika na kojoj se prikazuje tekst, snimka zaslona, Font

Opis je automatski generiran

***Programski kod TextView-a „radniPolozaj“***

**id –** označava id našeg TextView -a, u ovom slučaju to je „radniPolozaj“

**layout\_width –** određuje širinu našeg TextView -a; postavljen je na 0dp što znači da će TextView zauzeti onoliko mjesta koliko ima znakova

**layout\_height -** određuje visinu našeg TextView -a; postavljen je na wrap\_content što znači da će TextView zauzeti onoliko mjesta koliko mu je potrebno da obuhvati tekst

**layout\_weight –** težina elementa

**layout\_margin –** veličina sve 4 margine

**textColor –** boja teksta

**text –** tekst

**gravity -** poravnanje

**textSize –** veličina teksta

# **MainActivity**

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, računalo

Opis je automatski generiran

**lateinit var binding: ActivityMainBinding:** Ova linija deklarira svojstvo binding tipa ActivityMainBinding s odgodnim inicijaliziranjem (lateinit)

**private val database: DatabaseReference:** Ova linija deklarira privatno svojstvo database tipa DatabaseReference. DatabaseReference je klasa koja dolazi s Firebase Realtime Database bibliotekom. Ovdje se stvara instanca FirebaseDatabase s određenim URL-om za Firebase Realtime Database ("https://pmuprojekt-default-rtdb.europe-west1.firebasedatabase.app/"), a zatim se dobiva referenca na "zaposlenici" čvor u toj bazi podataka.

**binding = ActivityMainBinding.inflate(layoutInflater):** Ova linija koda koristi generiranu klasu ActivityMainBinding iz View Binding-a kako bi se inflatirao (stvorio) objekt binding.

**setContentView(binding.root):** Ova linija postavlja sadržaj aktivnosti na korijenski (root) pogled iz objekta binding. Metoda setContentView() se koristi u Android aktivnosti kako bi se postavio prikaz sadržaja aktivnosti.

**val ime = binding.ime.text.toString():** Ova linija dohvaća tekst unesen u polje s imenom "ime" pomoću View Binding-a i pretvara ga u string. Vrijednost se pohranjuje u varijablu ime.

**val prezime = binding.prezime.text.toString():** Slično kao prethodna linija, ova linija dohvaća tekst unesen u polje s imenom "prezime" i pretvara ga u string. Vrijednost se pohranjuje u varijablu prezime.

**val godine = binding.godine.text.toString():** Ova linija dohvaća tekst unesen u polje s imenom "godine" i pretvara ga u string. Vrijednost se pohranjuje u varijablu godine.

**val radniPolozaj = binding.radniPolozaj.text.toString():** Ova linija dohvaća tekst unesen u polje s imenom "radniPolozaj" i pretvara ga u string. Vrijednost se pohranjuje u varijablu radniPolozaj.

**var id = 0:** Inicijalizacija varijable id na 0. Ova varijabla će kasnije sadržavati jedinstveni identifikator za novog zaposlenika.

**if (!list.isEmpty()) id = list[list.size - 1].id + 1:** Ova linija provjerava je li lista list prazna. Ako nije prazna, dodjeljuje se vrijednost id zadnjeg zaposlenika u listi povećana za 1, čime se generira jedinstveni identifikator za novog zaposlenika.

**list.add(employees(ime, prezime, godine, radniPolozaj, id)):** Dodavanje novog zaposlenika u listu list. Koristi se konstruktor klase employees s podacima koje smo prikupili ranije.

**database.setValue(list**): Postavljanje vrijednosti u Firebase Realtime Database. Cijela lista list, koja sadrži podatke o zaposlenicima, postavlja se kao nova vrijednost u bazi podataka. Ova linija se koristi za ažuriranje podataka u stvarnom vremenu na Firebaseu.

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Multimedijski softver

Opis je automatski generiran

**list.clear():** Prije nego što se ažurira s novim podacima, stari podaci u listi list se brišu.

**try { ... }:** Pokušaj izvršiti sljedeći blok koda. U slučaju neuspjeha (iznimka), program će preskočiti blok i nastaviti s izvođenjem ostatka koda.

**val a: List<employees> = snapshot.children.map { ... }:** Pomoću snapshot.children prolazi kroz svu djecu (podatke) u trenutnom stanju baze podataka. Metoda map konvertira te podatke u listu objekata tipa employees.

**list.addAll(a):** Dohvaćeni podaci se dodaju u listu list.

**binding.recycler.apply { ... }:** Postavljanje RecyclerView-a pomoću View Binding-a s ažuriranom listom zaposlenika.

**adapter = employees\_adapter(list, this@MainActivity):** Postavljanje adaptera za RecyclerView. employees\_adapter je adapter koji se koristi za povezivanje podataka iz liste s prikazom u RecyclerView.

**onCancelled(error: DatabaseError):** Ova metoda se poziva ako čitanje iz baze podataka ne uspije, ali u navedenom kodu nije implementirana.

# **Data class „employees“**

Slika na kojoj se prikazuje tekst, snimka zaslona, Font, softver

Opis je automatski generiran

***Programski kod klase „employees“***

**data class employees(:** Ova linija označava početak definicije data klase imenom employees.

**var ime: String = "",:** Ovo definira svojstvo (property) ime tipa String unutar data klase employees. Zadana vrijednost je postavljena na prazan string.

**var prezime: String = "",:** Slično kao i prethodno, ova linija definira svojstvo prezime tipa String unutar data klase, s praznim stringom kao zadanim.

**var godine: String = "",:** Isto kao i prethodne dvije linije, ova linija definira svojstvo godine tipa String.

**var radniPolozaj: String = "",:** Ova linija definira svojstvo radniPolozaj tipa String.

**var id: Int = 0:** Ova linija definira svojstvo id tipa Int s zadanim vrijednostima postavljenim na 0.

# **Class „employees\_adapter“**

Slika na kojoj se prikazuje tekst, snimka zaslona, softver, Multimedijski softver

Opis je automatski generiran

***Programski kod klase „employees\_adapter“***

**val list: ArrayList<employees>,:** Ovo definira svojstvo list kao ArrayList objekata tipa employees. To je lista zaposlenika koju će adapter prikazivati.

**val th: Context:** Ovo definira svojstvo th kao objekt tipa Context. Context se obično koristi za pristup resursima, pozivanje aktivnosti i slično.

**): RecyclerView.Adapter<employees\_adapter.ViewHolder>() {:** Ova linija zatvara definiciju klase i označava da employees\_adapter nasljeđuje od RecyclerView.Adapter klase. Navodi se i da koristi ViewHolder unutar RecyclerView.Adapter.

**override fun onCreateViewHolder(:** Ova funkcija se koristi za stvaranje novog ViewHolder objekta.

**val v = ItemEmployeeBinding.inflate(LayoutInflater.from(th), parent, false): Ovdje** se inflatira (stvara) objekt ItemEmployeeBinding iz layouta.

**return ViewHolder(v):** Vraća se novi ViewHolder stvoren s ItemEmployeeBinding objektom.

**override fun onBindViewHolder(:** Ova funkcija se koristi za postavljanje podataka iz list na određenu poziciju position u ViewHolder-u.

**holder.bindItem(list[position], th):** Poziva se funkcija bindItem na ViewHolder objektu, a kao argumenti se predaju podaci (zaposlenik) na toj poziciji i Context objekt.

**override fun getItemCount(): Int {:** Ova funkcija vraća ukupan broj stavki u listi. To je važno za RecyclerView kako bi znao koliko stavki treba prikazati.

**class ViewHolder(:** Ovo označava početak definicije unutarnje klase ViewHolder. ViewHolder je odgovoran za držanje referenci na elemente sučelja koje je potrebno ažurirati.

**private var itemBinding: ItemEmployeeBinding:** Ovo definira svojstvo itemBinding tipa ItemEmployeeBinding, koje drži referencu na View Binding objekt za svaki element retka.

**RecyclerView.ViewHolder(itemBinding.root):** Poziva konstruktor roditeljske klase i prosljeđuje mu root View iz ItemEmployeeBinding.

**fun bindItem(tekst:employees, th:Context){:** Ova funkcija se koristi za postavljanje podataka (zaposlenika) na odgovarajuće elemente sučelja.

**itemBinding.ime.text = tekst.ime:** Postavljanje teksta za ime zaposlenika na odgovarajući element sučelja.

**itemBinding.prezime.text = tekst.prezime:** Postavljanje teksta za prezime zaposlenika na odgovarajući element sučelja.

**itemBinding.godine.text = tekst.godine:** Postavljanje teksta za godine zaposlenika na odgovarajući element sučelja.

**itemBinding.radniPolozaj.text = tekst.radniPolozaj:** Postavljanje teksta za radni položaj zaposlenika na odgovarajući element sučelja.

**itemBinding.id.text = tekst.id.toString():** Postavljanje teksta za identifikator zaposlenika na odgovarajući element sučelja, pretvarajući id u string.