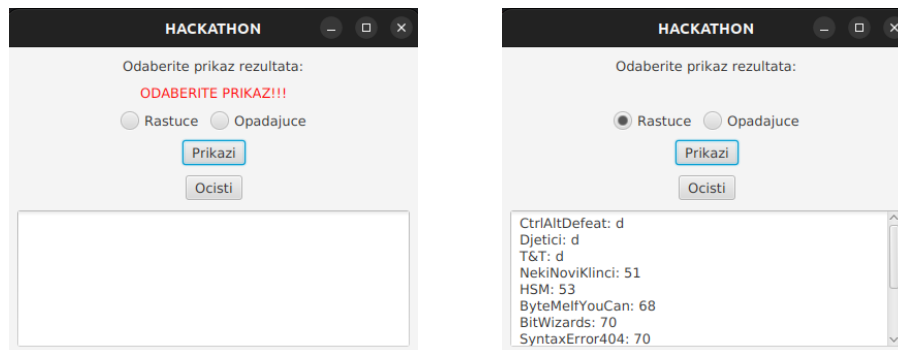


# Objektno-orijentisano programiranje, primer ispita 2023/24.

1. Implementirati sistem koji omogućava rad sa geometrijskim figurama (krugovi, kvadrati i pravougaonici).
  - (a) Definirati apstraktnu klasu *GeometrijskaFigura* koja ima:
    - Atribut *ime* koje predstavlja naziv geometrijske figure.
    - Apstraktnu metodu *povrsina()* koja će biti implementirana u svakoj konkretnoj geometrijskoj figuri.
  - (b) Implementirati tri konkretne klase koje nasleđuju apstraktnu klasu *GeometrijskaFigura*:
    - *Krug* sa dodatnim atributom *poluprecnik*.
    - *Kvadrat* sa dodatnim atributom *duzinaStranice*.
    - *Pravougaonik* sa dodatnim atributima *duzina* i *sirina*.
  - (c) Definirati interfejs *Ispis* koji ima apstraktnu metodu *prikazi()* koja će biti implementirana u svakoj klasi. Metoda *prikazi()* treba da ispisuje informacije o geometrijskoj figuri.
  - (d) Implementirati klasu *GeometrijskiSkup* koja sadrži niz objekata tipa *GeometrijskaFigura* i definirati konstruktor koji kao argument dobija niz figura. U klasu dodati metodu *ukupnaPovrsina()* koja vraća sumu površina svih figura u skupu.
  - (e) U klasi *Main* napisati kod koji koristi ove klase i interfejse:
    - Kreirati instancu klase *GeometrijskiSkup*.
    - Dodati nekoliko instanci geometrijskih figura različitih tipova (krug, kvadrat, pravougaonik) u skup.
    - Pozvati metodu *ukupnaPovrsina()* i ispisati rezultat.
    - Iterirati kroz skup i za svaku figuru pozvati metodu *prikazi()*.
2. Napisati aplikaciju korišćenjem JavaFX biblioteke koja izgleda kao na slikama. U datoteci *Rezultati.txt* koja se nalazi u korenom direktorijumu projekta nalaze se rezultati takmičenja. Sadržaj reda datoteke je naziv tima (jedna reč) kao i broj poena koje je tim osvojio ili karakter *d* koji označava da je tim diskvalifikovan. Klikom na dugme *Prikazi* u zavisnosti od označenog radio dugmića u tekstualnom polju se prikazuje rezultat u rastućem, odnosno opadajućem poretku. Klikom na dugme *Ocisti* tekstualni sadržaj se briše kao i tekst iz labela za grešku (ukoliko postoji) i označenja sa radio dugmića se uklanjaju.



3. Implementirati sistem koji služi za pronalaženje maksimuma u kolekciji.
  - (a) Implementirati generički interfejs *GenericMax* koji sadrži metode:
    - `Optional<T> findMax(Collection<T> collection, Comparator<T> comparator);`
    - `Optional<T> findMax(Collection<T> collection);`
  - (b) Implementirati klasu *MaxFinder* koja može ili ne mora pri pravljenju objekta da primi comparator objekat. Klasa treba da implementira interfejs *GenericMax*. Drugi metod iz interfejsa bi trebalo da koristi comparator ukoliko je prosleđen prilikom konstrukcije objekta, inače obraditi grešku.
  - (c) U klasi *Main* napraviti kolekciju klase *Tacka* i u kolekciju ubaciti nekoliko tačaka.
    - Testirati metodu *findMax* sa dva argumenta koristeći komparator tačaka po *x* koordinati rastuće (ako su *x* koordinate jednake porediti po *y* rastuće).
    - Testirati metodu *findMax* sa jednim argumentom koristeći komparator tačaka po *y* koordinati opadajuće (ako su *y* koordinate jednake porediti po *x* opadajuće).Ukoliko maksimum (u odgovarajućem poretku) postoji, ispisati ga.