1.Što je JPF?
Radni okvir i skup alata.

2. Za što ?
Formalnu verifikaciju

3. Čega?
Programa pisanih u Javi

4. Kojom metodom
Metodom provjere modela

5. Što je jezgra JPF-a
VM koji se izvodi iznad JVM

6. Zašto je razvijen JPF?
automatsko kodiranje UML + analiza dosezljivosti svih dijelova koda

7. Ako su dijelovi koda predstavljeni kao stanja programa onda se govori o
Analizi dosezljivosti (reachability analysis)

8. Nad čime JPF izvodi provjeru modela?
SUT system under test (bajtkod java aplikacije koja se treba provjeriti)

9. SUT aplikacije mogu biti napisane na 3 načina, koja su to poredano po učestalosti:
1) Neovisno o JPF-u
2) S podrškom za JPF
3) Ovisno o JPF-u

10. Kako se specificiraju svojstva koja SUT treba zadovoljiti?
Skupom konfiguracijskih datoteka

11. Što su ulazi u JPF a što izlaz?
Ulazi su bajtkod SUT-a i konfiguracijske datoteke a izlaz izvještaj provjere.

12. **Za što se koristi JPF?**
    1)Istraživanje alternativnih izvršavanja
        Sekvence raspoređivanja
        Varijacije u ulaznim podatcima
        Događaji okoline
        Izbori kontrolnog toka
    2)Inspekcija izvršavanja (pokrivnost koda, preljev, podljev)

13. Kako JPF izvršava naš SUT?
Na sve moguće načine

14. Što napravi ako nađe mjesto kvara?
Izvještava o svakom koraku od početka do mjesta kvara.

15. Kako JPF pamti stanja i što to znači?
Eksplicitno, što znači da pamti stog ,heap i stanja dretvi

17. Eksplicitno pamćenje stanja dovodi do kojeg problema?
Eksplozije broja stanja

18. Čemu služi class loader subsystem JVM-a?
Učitava razrede i sučelja preko njihovog punog imena

19. Što su runtime data areas JVM-a?
Memorijski prostori koje organizira JVM

20. **Navedi runtime data areas**?

Zajedničko svim dretvama:
1)**Method area** - inf o metodama i atributima razreda
2)**Heap** - živi objekti

Svaka dretva ima svoje:
3)**Java stacks** - novi stack frame za svaki poziv metode
4)**PC register** - programska brojila
5)**Native method stack** - zasebni stogovi za izvršavanje nativnih metoda

21. **Koja mogu biti stanja dretvi**

1) NEW - instancirana, nepokrenuta
2) RUNNABLE - pokrenuta
3) BLOCKED - čeka na monitor
4) WAITING - čeka neodređeno dugo na drugu dretvu da završi s nečim
5) TIMED_WAITING čeka određeno vrijeme na drugu dretvu
6) TERMINATED - završila izvođenje svog run-a

22. Gdje se može navesti Javina ključna riječ synchronized, čemu služi?
Iznad metode ili bloka naredbi, služi za modeliranje kritičnog odsječka

23. Na koja 3 načina se rješava problem eksplozije broja stanja:

1) Provjera podudaranja stanja (backtrackanje na neistraženo)
2) Djelomično smanjenje poretka (grupiranje nizova instrukcija koje ne uzrokuju efekte van jedne dretve)
3) Delegiranje izvođenja JVM-u (samo onih stvari koje ne utječu na verifikaciju)

24. **Od kojih 6 podfoldera se sastoji direktorij src JPF-a**

1. Main - glavni razredi (izvođenje na host JVM)
2. Peers - paketi s ostalim razredima koji se izvode umjesto pravih nativnih metoda (JVM)
3. Classes - razredi koji služe kao biblioteke JPF-a za našu aplikaciju (VM JPF-a)
4. Annotations
5. Examples
6. Tests

25. Kako je organiziran podfolder build?

Isto samo sadrži .class datoteke

26. Koji se sustav koristi za buildanje JPF-a?

Ant

27. Koji razredi su glavni u poddirektoriju Main JPF-a?

JVM (proizvodi stanja)
Search (pretražuje ih)

28. Za specifikaciju i provjeru svojstava treba nam kakav konfiguracijski objekt?

1) Zasnovan na stringu
2) Po volji proširiv
3) Prenosiv odozgo nadolje u hijerarhijskom procesu (svaka komponenta uzima sta joj treba)

29. Sam konfiguracijski objekt ostvaren je kojim razredom?

`gov.nasa.jpf.Config`

30. Konfiguracija se može ostvariti na koje 4 razine (niža nadjačava višu):

1. Cjelokupna instalacija (parametri u `<user_home>/.jpf/site.properties`)
2. Projektna instalacija (`npr. jpf-aprop/jpf.properties`) projekti JPF-a
3. SUT (npr. `DummyProject/src/HelloWorld.jpf (uz HelloWorld.java)`)
4. Komandno linijski argumenti (navođenjem `<ključ>+=vrijednost`)

31. Mora li se jpf fajl zvati isto kao java fajl kojeg provjerava (unutar SUT-a).

Ne, jer se to specificira unutar jpf file-a navođenjem razreda kojeg treba provjeriti linijom:

```
target = <ImeRazreda>
```

32. Navedi tri razreda za izvještaje

1)Reporter -
2)Publisher - proizvodi izlaz ovisno o željenom obliku (tekst, XML)
`ConsolePublisher`
3)PublisherExtension - spec izdavači za određena svojstva

33. Teme izvještavanja faze propertyViolation:

1)Error - tip i detalji kršenja svojstva koje je pronađeno
2)Trace
3)Snapshot - lista stanja svake dretve
4)Output

34. Faza finish ima po defaultu određene teme:

1)Results - je li došlo do kršenja svojstava
2)Statistics - ukupna statistika izvođenja JPF-a na SUT-u

35. Proširenja JPF-a su

1) Listeneri - observeri koji reagiraju na određene događaje prilikom pretrage ili rada JVM-a
2) Generatori izbora - sustavno istraživanje prostora stanja
3) Provjera anotacijskih svojstava

36. Koje Adaptere nasljeđuju koji listeneri?

1. `ListenerAdapter - CoverageAnalizer, DeadlockAnalizer, NullTracker`
2. `PropertyListenerAdapter - PreciseRaceDetector, NoStateCycles`

37. Koje zadatke pokrivaju anotacijska svojstva?

1. Nedozvoljeno dodjeljivanje null-a (@Nonnull)
2. Ugovore (@Requires, @Ensures, @Invariant)
3. Promjenjivost objekta (@Const)