

JavaPathFinder

- radni okvir i skup alata za FV programa pisanih u Javi, koristi **metodu provjere modela**
- jezgra JPF-a = VM koji se izvodi iznad JVM-a
- zašto – automatsko kodiranje UML dijagrama stanja u Java program i analiza dosegljivosti svih dijelova koda (predstavljeni kao stanja)
- radi provjeru nad kompajliranim Java aplikacijama – **tzv. SUT (System Under test)**
 - o ulaz: bytecode, izlaz: izvještaj (report)
- specificiranje svojstava: konfiguracijske datoteke
- pisanje Java prog.: neovisno o JPF-u, s podrškom za JPF (imaju anotacije JPF-a), ovisno o JPF
- nije namijenjen za sekvencijske aplikacije s malim brojem ulaznim vrijednosti (nepotreban)
- koristi se za:
 - o **istraživanje alternativnih izvršavanja**
 - sekvence raspoređivanja - konkurentne aplikacije, varijacije u ulaznim pod., reakcije na događaje okoline, izbor kontrolnog toka (sustavno pretraživanje)
 - o **inspekciju izvršavanja programa**
 - analizador prekrivanja koda ili neinvazivnih testova uvjeta koje je teško otkriti
- otkriva:
 - o jezgra JPF-a (**jpf-core**) – nefunkcijska svojstva – koja se ne smiju nikad dogoditi
 - potpuni zastoj, utrke, neuhvaćene iznimke, assert provjere
 - o proširenja JPF-a – korisnička svojstva, uglavnom koriste **slušače (listenere)**
 - o kompletnu povijest izvršavanja (**trace**) – do razine bytecodea
- princip rada:
 - o VM koji SUT izvodi na sve moguće načine, kod kvara izvještava o svakom koraku
 - o Eksplicitno pamćenje stanja izvođenja programa u jpf-core
 - Okvir stoga dretvi, objekte na gomili, stanje dretvi
 - ➔ **Problem eksplozije broja stanja**
 - o JVM iznutra
 - Class files – bytecode, class loader – mehanizam učitavanja tipova s punim kvalificirajućim imenom
 - Runtime data areas – memorijski prostor
 - Method area – metode, atr. i konstante razreda
 - Heap – memorijski prostor za objekte
 - Java stacks i PC register - dobije ih svaka dretva
 - Native method stack – stog za dretve koje izvršavaju metode ovisne o OS-u
 - Execution engine – mehanizam koji izvršava međukod
 - o Stanja dretvi: **new** – instancirana, nepokrenuta, **runnable** – izvršava se, **blocked** – čeka k.o., **waiting** – neodređeno dugo čeka drugu dretvu, **timed waiting** – određeno dugo čeka, **terminated** – završila je svoju metodu run
 - o Rješavanje eksplozije stanja – provjera podudaranja stanja, djelomično smanjenje poretka – različiti poretki dovode do istog stanja, delegiranje izvođenja metoda JVM-u – kad nemaju utjecaja na verifikaciju
- k.r. synchronized – ispred metode ili bloka naredbi ili objekta sinkronizacije

Struktura JPF-a

- **src** folder JPF-a sadrži foldere: **main** – razredi ključni za jpf-core (izvode se na host JVM-u), **peers** – paketi s razredima čije se metode izvode umjesto nativnih (na host JVM-u), **classes** – razredi koji se

izvršavaju na JPF VM-u, **annotations** – javine anotacije koje JPF treba procesirati, **examples** – primjeri SUT-ova, **tests** – tipični ispitni slučajevi

- **build** folder sadrži kompajlirane dijelove src-a, JPF se pokreće uz **RunJPF.jar**
- nbproject/eclipse – netbeans/eclipse folderi, lib – dodatne knjižnice, doc – tekstualna dokument.
- **Main** – jezgra jpf-corea, najbitnije klase VM – proizvodi stanja i Search – upravljački program

Specifikacija i provjera svojstava programa

- Svojstva se navode u metodi koja ih treba provjeriti (Nonnull, Deadlock, ...)
- Konfiguracijski dokument je fleksibilan, baza: stringovi, put: **gov.nasa.jpf.Config**
- 4 razine konfiguracije (od više do niže, niže nadjačavaju više):
 - o cjelokupna instalacija na računalu
 - datoteka **site.properties**, key=value zapis, mora postojati, traži objekte JPF-a
 - o projektne instalacije – za svaku komponentu JPF-a
 - svaki projekt JPF-a u korijenskom folderu ima **jpf.properties**
 - o naše aplikacije (SUT)
 - u datoteci ***.jpf**, mora se navesti ciljani glavni razred **target = MainClass**
 - o komandno-linijski argumenti
- Izvještaji
 - o Razred izvjestitelj **Reporter**, razredi izdavači **Publisher** te njihovi proširitelji **PublisherExtensions**
 - o Faze: start, property_violation, finish
 - Property_violation: error, trace, snapshot (stanje dretvi), output (ispis programa)
 - Finish: results, statistics
 - o Uobičajen izdavač: ConsolePublisher

Napredne teme i proširenja JPF-a

- Slušači
 - o Najvažniji mehanizam proširenja, Observeri koji reaguju na događaj prilikom pretraže (SearchListener) ili rada JVM-a (VMLListener)
 - o Navode se svojstva listener u projektu ili nižim razinama
 - o Konkretni nasljeđuju razred **Adapter** koji ima prazne impl. Metode Search/VMListenera
 - o **ListenerAdapter** služi za skupljanje informacija o JPF-u pa ga koriste **CoverageAnalyzer**, **DeadlockAnalyzer**, **NullTracker**
 - o **PropertyListenerAdapter** se koristi za svojstvo programa npr. **NoStateCycles**, **PreciseRaceDetector**
- Generatori izbora
 - o Cilj: doći do zanimljivih situacija
 - o Mogućnosti: raspoređivanje dretvi, vrijednosti podataka, kontrolni tok
 - o Heuristike pretraživanja – odstupa se od potpunog pretraživanja
 - o **ChoiceGenerator**, **ThreadChoiceGenerator**
- Provjera anotacijskih svojstava – projekt **jpf-aprop**
 - o Navode se slušači u *.jpf te importira jpf-aprop-annotations.jar
 - o Provjeravaju:
 - @Nonnull
 - ugovore @Requires, @Ensures, @Invariant
 - obilježja sigurnosti pristupa polju rareda @GuardedBy
 - promjenjivost objekta @Const