



IntelliJ IDEA - dodaci koje morate imati

1. Pregled

IntelliJ IDEA razvojno okruženje najpopularnije je okruženje kada govorimo o onima za razvoj Java aplikacija. Kao što smo naveli u objavi [Od kuda krenuti](#), funkcionalnosti IntelliJ razvojnog okruženja moguće je proširiti pomoću dodataka (engl. plugins). Ovaj članak donosi Vam popis dodataka IntelliJ razvojnog okruženja koji će Vam olakšati programiranje.

2. Jednostavniji razvoj



IntelliJ IDEA razvojno okruženje dolazi s mnogo funkcionalnosti koje će Vam olakšati (i ubrzati) razvoj aplikacija, čak i bez instalacije dodatka. Od predložaka za automatsko popunjavanje koda, prečaca pomoću kojih je moguće izvoditi sve akcije u okruženju bez korištenja miša, pa sve do jednostavnijeg pokretanja, pakiranja i debugiranja aplikacija (naravno, i za navedene akcije postoji prečac na tipkovnici). Funkcionalnosti je zaista pregršt.

Naravno, uvijek postoje dodaci koje možete instalirati kako bi Vam razvoj aplikacija bio još jednostavniji. U nastavku slijede neki od njih.

2.1. Key Promoter X

[Key Promoter X](#) je definitivno jedan od dodataka kojeg želite imati u svom razvojnom okruženju. Ovaj dodatak pomaže Vam naučiti IntelliJ prečace na tipkovnici (engl. shortcuts) u trenutku samog programiranja. Kada izvršite određenu radnju za koju postoji shortcut, u okruženju će se prikazati skočni prozor u kojem će biti navedena jedna ili kombinacija tipaka pomoću kojih se navedena radnja može izvršiti. Korištenjem prečaca ubrzat ćete svoje programiranje jer neće biti potrebe za korištenjem miša.

Plugin za IntelliJ inačice: Ultimate, Community

2.2. Java decompiler

Pomoću [Java Decompiler](#) dodatka možete otvoriti bilo koju kompajliranju klasu (datoteku s .class ekstenzijom). Bez korištenja ovog plugina, otvaranjem .class datoteke vidjeli biste gomilu znakova koje ne biste mogli razumjeti.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

2.3. Save Actions

Želite definirati set radnji koje će se izvršiti prilikom spremanja (poput formatiranja i optimizacije koda)? Ako je odgovor da, onda je dodatak [Save Actions](#) pravi izbor za Vas. Akcije koje možete konfigurirati su:

- optimizacija import izjava
- reformatiranje koda
- kompajliranje koda
- razmještaj koda unutar klase

Važno je napomenuti da je moguće konfigurirati u kojem trenutku će se navedene akcije dogoditi – pritiskom na gumb Save ili pri automatskom spremanju, odnosno sinkronizaciji.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

2.4. Maven Helper

[Maven Helper](#) dodatak koristit će Vam ako koristite Maven razvojni alat (engl. building tool). Pomoću ovog dodatka možete na jednostavan način analizirati ovisnosti (engl. dependencies) dodane u projekt i, po potrebi, izostaviti (engl. exclude) one koji su u konfliktu. Jednostavnije možete pokretati, odnosno debugirati module unutar aplikacije.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

2.5. GenerateAllSetter

[Generate All Setter](#) dodatak je koji će Vam koristiti prilikom postavljanja vrijednosti nekom objektu korištenjem setter metoda. Nećete se više morati brinuti i provjeravati jesu li postavili sve vrijednosti korištenjem setter metoda. Nakon što kreirate novi objekt, pomoću navedenog plugina možete jednostavno pozvati sve setter metode (uključujući i setter metode nadklase).

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

3. Clean code



Imati “?isti k?d” danas je postao standard. Martin C. Robert, rekao je: “It is not enough for code to work.”. Danas nije dovoljno napisati program koji ?e obavljati odre?enu zada?u, ve? je važno obratiti pozornost i kako je isti napisan.

Ako radite kao programer, sigurno ste se našli u situaciji u kojoj ste morali raditi na aplikaciji koju je napisao drugi programer, bilo da ste dodavali novu funkcionalnost ili ispravljali postoje?e. Jeste li bili sretni rješenjem koje je prethodni programer ostavio za sobom ili ste prolazili kroz kod pokušavaju?i pohvatati dretve koje su se zbivale u glavi programera koji je pisao kod? Nadamo se da je ovo prvo.

Nema ništa gore od situacije kada prolazite kroz tu?i k?d pokušavaju?i otkriti što odre?eni dio k?da radi. Možda ?ete biti ljutiti što morate raditi na takvoj aplikaciji. S druge strane, važno je zapitati se ?inite li i sami sve što možete kako biste iza sebe ostavili lako razumljiv k?d?

U prošlosti je možda bilo jedino važno da program radi ispravno. Danas to više nije tako. Prilikom razvijanja aplikacije, trebali biste obratiti pozornost i na na?in na koji pišete kod. U nastavku slijede dodaci koji Vam mogu pomo?i oko održavanja “?istog k?da”.

3.1. SonarLint

[SonarLint](#) može Vam uvelike pomo?i oko poboljšanja kvalitete k?da. Baš poput spell checkera, ovaj dodatak otkriva nedostatke k?da i daje Vam sugestije odmah, u trenutku pisanja, odnosno razvijanja aplikacije. Svaka sugestija popra?ena je objašnjenjem zašto je jedna opcija bolja od druge. Ponekada, ako ste u žurbi, može Vam se prikristi poneka greškica, zato je tu SonarLint – njemu nikada ništa ne promakne.

Plugin je dostupan za IntelliJ ina?ice: Ultimate, Community

3.2. SpotBugs

[SpotBugs](#) dodatak donosi mogućnost analize k?da s namjerom pronalaska mogućih pogrešaka (engl. bugs)

) unutar istog. Ovaj dodatak može otkriti više od 400 situacija koje mogu uzrokovati pogreške unutar koda. Može pomoći pri lociranju dijelova koda koji mogu prouzrokovati NullPointerException iznimku, otkrivanja beskonačnih i rekurzivnih petlji te lošeg korištenja Java biblioteke. Analizom ćete dobiti listu potencijalnih grešaka uz objašnjenje zašto bi određena linija koda mogla uzrokovati problem.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

3.3. QAPlug – FindBugs

Još jedan od odličnih alata koji može poslužiti kao pomoć pri pronalasku potencijalnih bugova u kodu. Da biste mogli koristiti [QAPlug – FindBugs](#) dodatak, morate instalirati [QAPlug](#) dodatak. QAPlug je dodatak IntelliJ IDEA za upravljanje kvalitetom koda koji integrira alate kao što su PMD, Checkstyle i Findbugs.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

3.4. Design Patterns IntelliJ IDEA Plugin

[Design Patterns](#) dodatak omogućava implementaciju design patterna unutar koda. Implementirati Builder pattern nikada nije bilo jednostavnije! Par klikova miša i voilà. Ipak, navedeni dodatak ne podržava svih 23 design patterna. To je, naravno, razumljivo obzirom na kompleksnost koja stoji iza nekih od njih. Podržava implementaciju sljedećih obrazaca: Strategy, Builder, Factory, Singleton te strukturalne obrasce.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

4. Ostali dodaci



U nastavku slijede još neki od dodataka koji bi Vam mogli koristiti.

4.1. GitToolBox

[GitToolBox](#) donosi Vam proširene funkcionalnosti već postojeće integracije s Gitom. Pomoću ovog plugina možete vidjeti tko je commitao i kada određenu liniju koda, automatski dohvaćati kod, vidjeti koja linija je commitana, a koja nije. Također, možete jednostavnije commitati Vaše promjene.

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

4.2. Nyan Progress Bar

[Nyan Progress Bar](#) je za sve Nyan catoljupce. Ako želite razbiti monotoniju IntelliJ razvojnog okruženja, ovaj (ili slični dodaci) mogu Vam pomoći u tome. Poznati Java Champion, Victor Rentea, također koristi ovaj plugin. Ono što ćete dobiti ovim pluginom jest progress bar s nyan cat ikonom.

Plugin za IntelliJ inačice: Ultimate, Community

4.3. Statistic

Zanima Vas od koliko linija kada se sastoji Vaš projekt? Koliko linija kada ima svaka klasa? Koja je veličina datoteka unutar projekta? Ako je odgovor da, [Statistic](#) dodatak Vam to omogućava! Prikazuje statistiku projekta, odnosno datoteka razvrstanih prema njihovom proširenju zajedno s veličinom, brojem redaka, brojem komentara i slično. Sami možete odabrati opseg za koji želite generirati statistiku (projekt, modul, paket ili datoteka).

Plugin je dostupan za IntelliJ inačice: Ultimate, Community

5. Zaključak

Dodaci navedeni u ovoj objavi mogu Vam koristiti bez obzira na posao kojeg obavljate. Ako ste početnik, možda biste htjeli instalirati dodatke tek kada razumijete gradivo. Bez obzira što Vam dodaci, poput GenerateAllSetters ili Design Patterns, olakšavaju razvoj, trebali biste znati napisati kod kojeg oni generiraju i bez njihovog korištenja. Tek kada razumijete kako napisati, primjerice Builder Pattern, i kako funkcionira, trebali biste se prebaciti na korištenje dodatka.

[Read More](#)
