Java 18 update

Spis treści

Java 18 update

Java 18 została wydana w marcu 2022 i jest wersją **non-LTS**. Poniżej omówimy niektóre funkcjonalności udostępnione w tym wydaniu. Przy aktualizacji wersji Javy często poprawianych jest o wiele więcej funkcjonalności i dodawanych o wiele więcej klas lub metod niż te, które wymieniamy tutaj. W obrębie tych materiałów poruszamy tylko te kwestie, które są adekwatne do naszego poziomu zaawansowania jako Java developerów.

Metoda finalize() deprecated

Pamiętasz metodę `finalize()`? Garbage Collector wołał tę metodę przez zwolnieniem pamięci, którą zajmuje obiekt. Metoda ta mogła być wykorzystana np. żeby zwolnić zasoby przed usunięciem obiektu. Przykładowo moglibyśmy dodać implementację tej metody, w której zamykamy wszystkie otwarte pliki. Nigdy wcześniej w materiałach nie implementowaliśmy tej metody, bo przykładowo otwarte pliki albo zasoby zamykaliśmy automatycznie w inny sposób. Stosowanie metody `finalize()` miało swoje wady:

- Nie ma gwarancji, kiedy zostanie wywołane GC, zatem opieranie się na tej metodzie może oznaczać, że będziemy musieli długo poczekać, zanim jakieś zasoby zostaną zwolnione.
- Metoda finalize jest wołana dla każdej instancji klasy, czyli dla każdego usuwanego obiektu. Nie ma gotowego mechanizmu, który pozwoli nam wybrać obiekty, dla których `finalize()` ma zostać wywołane, a dla których nie.
- Różne źródła mówią, żeby nie wywoływać tej metody ręcznie, a często nawet jej nie implementować, bo nie ma żadnej gwarancji, że ta metoda w ogóle zostanie wywołana.

Między innymi z tych, jak i z innych powodów, metoda `finalize()` została w Java 9 oznaczona jako `@Deprecated`. W Java 18 metoda ta została oznaczona jako `Deprecated for removal`.

Po co o tym wspominamy? Może się kiedyś przyda jako ciekawostka na rozmowach rekrutacyjnych. 😉

Pattern Matching for switch (preview)

Mechanizm **Pattern Matching for switch** doczekał się kolejnych zmian i usprawnień, natomiast nadal został utrzymany jako **preview feature**. Mechanizm ten zostanie omówiony w pełni, gdy zostanie udostępniony jako **standard feature**.

Podsumowanie

Dlaczego tak szybko skończyliśmy? Java 18 wprowadziła relatywnie mało zmian. Większość zmian wprowadzonych w Javie 18 dotyczyła mechanizmów, których albo nie umiemy, albo są zbyt skomplikowane na tym poziomie nauki, albo twórcy Zajavki uznali, że z naszego punktu widzenia zmiany te nie są aż tak istotne i lepiej poświęcić ten sam czas na skupienie się na dalszych zagadnieniach. Dlatego właśnie zostało omówione tak mało zagadnień. ☺

Przypomnę, że przy aktualizacji wersji Javy często poprawianych jest o wiele więcej funkcjonalności i dodawanych o wiele więcej klas lub metod niż te, które wymieniamy tutaj. Z kolejnymi wersjami wprowadzane są również rozmaite poprawki lub usprawnienia w samym działaniu JVM albo przykładowo Garbage Collectora (w tym przypadku mogą to być, chociażby różne algorytmy, o których działanie oparty jest GC). Zmianom mogą ulegać również kwestie dotyczące zarządzania pamięcią. Oprócz tego kolejne wersje Javy mogą również wprowadzać dodatkowe narzędzia, które programista może wykorzystywać w swojej pracy. Do tego poprawkom mogą podlegać istniejące implementacje metod. W obrębie tych materiałów poruszamy tylko te kwestie, które są adekwatne do naszego poziomu zaawansowania jako Java developerów.

Jeżeli natomiast interesuje Cię, jakie jeszcze zmiany są wprowadzane z każdą wersją — wystarczy, że wpiszesz w Google np. "Java 18 features" i znajdziesz dużo artykułów opisujących wprowadzone zmiany. Możesz również zerknąć na tę stronę JDK 18. Zaznaczam jednak, że wiele funkcjonalności będzie niezrozumiałych. ☺