

Czego się dowiesz

- Jakie skróty klawiszowe warto zapamiętać w IntelliJ IDEA,
- jakie dodatkowe funkcjonalności oferuje.

Wprowadzenie

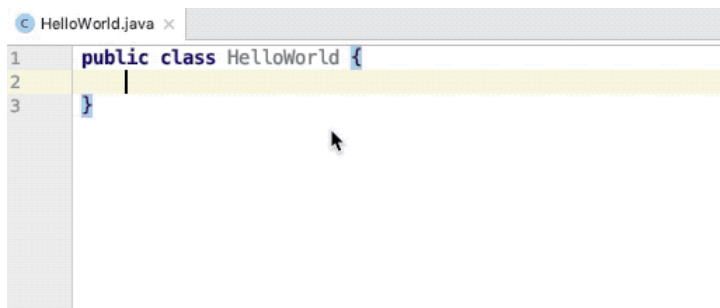
IntelliJ IDEA jest środowiskiem programistycznym, które mocno promuje programowanie bez używania myszy, czy gładzika, a zamiast tego stawia na intensywne korzystanie z klawiatury i skrótów. W większości przypadków wystarczy ci znajomość tylko kilku skrótów, jednak z czasem warto zainteresować się bardziej zaawansowanymi możliwościami, które dodatkowo podniosą naszą efektywność.

Autouzupełnianie kodu

Jeśli jeszcze tego nie potrafisz, to warto poćwiczyć naukę bezwzrokowego pisania. Jeśli skupisz się na tym co pojawia się w IntelliJ to zauważysz, że w wielu przypadkach wystarczy zapisanie tylko kilku pierwszych liter kodu, a resztę IntelliJ wygeneruje automatycznie. Kilka przykładów poniżej.

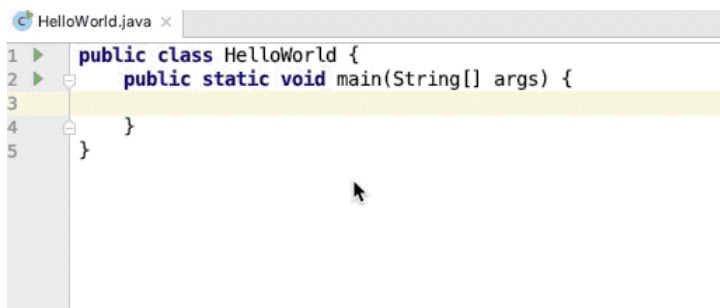
Każda aplikacja potrzebuje klasy z metodą startową main. Wystarczy, że po utworzeniu klasy wpiszesz ciąg znaków **psvm** (jak **public static void main**), a IntelliJ wygeneruje szablon metody main.

W podobny sposób możesz wygenerować metodę `System.out.println()`. Wystarczy, że wpiszesz **sout** i wciśniesz Enter.



```
HelloWorld.java x
1 public class HelloWorld {
2
3 }
```

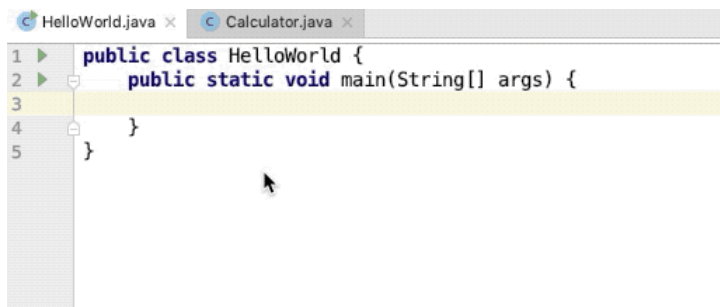
Jeżeli podpowiedź w IntelliJ zniknie, możesz ją przywrócić korzystając ze skrótu **Ctrl + Spacja**.



```
HelloWorld.java x
1 public class HelloWorld {
2 public static void main(String[] args) {
3
4 }
5 }
```

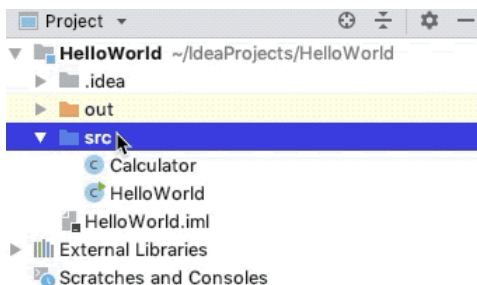
Podstawowe skróty klawiaturowe

Alt + Enter lub **⌘ + Enter** - podpowiedzi kontekstowe. Przykładowo możesz najpierw wywołać metodę na jakimś obiekcie, a IntelliJ podpowie ci, że wynik tej metody można przypisać do zmiennej.

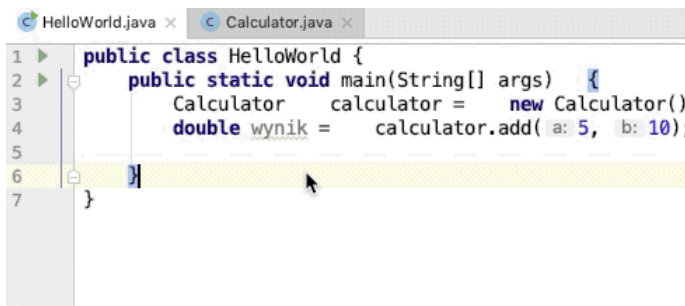


```
HelloWorld.java x Calculator.java x
1 public class HelloWorld {
2 public static void main(String[] args) {
3
4 }
5 }
```

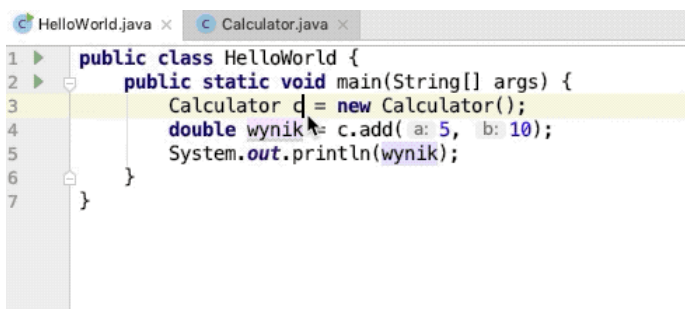
Alt + Insert lub **⌘ + N** - tworzenie nowych klas lub innych plików.



Ctrl + Alt + L lub ⌘ + ⌘ + L - automatycznie formatowanie kodu.



Shift + F6 - zmiana nazwy zmiennej/klasz/metody - automatycznie poprawione zostaną wszystkie wystąpienia danego elementu.



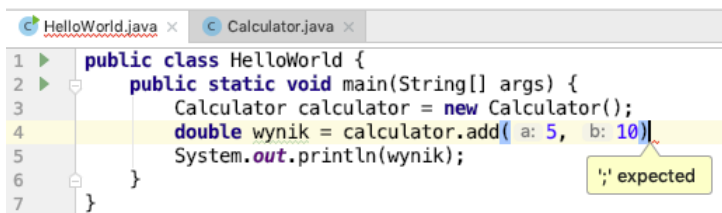
Inne:

- Podwójne wciśnięcie Shift - opcja "Szukaj wszędzie"
- Alt + 1 lub ⌘ + 1 - przejście do drzewa projektu
- Esc - przejście do okna edytora

Na początek powyższe skróty powinny być wystarczające do komfortowej pracy. Polecam ćwiczenie polegające na tym, żeby jak najrzadziej korzystać z myszy lub touchpada.

Wykrywanie błędów i sugestie

IntelliJ IDEA na bieżąco weryfikuje nasz kod pod kątem składniowym. Jeżeli w jakimś miejscu zapomnimy np. o średniku, nie domkniemy klamry, to będziemy o tym wiedzieli już przed próbą uruchomienia programu. IntelliJ podświetli takie fragmenty kodu na czerwono i jest to sygnał, że trzeba coś naprawić. Po najechaniu kursorem na takie miejsce wyświetlony zostanie dymek z informacją co jest nie tak. Poniżej przykład kodu, w którym zapomniałem dodać średnika na końcu wiersza.



Czasami IntelliJ będzie podświetlał fragment kodu na żółto lub szaro. Są to sugestie co do naszego kodu. Przykładowo jeżeli w programie zdefiniujemy zmienną, ale do niczego jej nie wykorzystamy, to IntelliJ zmieni kolor takiej zmiennej na szary informując nas o tym, że jest ona nieużywana. W takiej sytuacji po wciśnięciu kombinacji Alt + Enter zasugeruje nam jej usunięcie.

```
HelloWorld.java x Calculator.java x
1 public class HelloWorld {
2     public static void main(String[] args) {
3         Calculator calculator = new Calculator();
4         double wynik = calculator.add( a: 5, b: 10);
5     }
6 }
7 }
```