


Zintegrowane środowisko programistyczne

- [Wprowadzenie](#)
- [Przeczytaj](#)
- [Prezentacja multimedialna](#)
- [Sprawdź się](#)
- [Dla nauczyciela](#)



Zintegrowane środowisko programistyczne

Źródło: Thomas Tastet, domena publiczna.

W tym e-materiale powtarzamy wiadomości ze szkoły podstawowej.

W pracy programisty bardzo istotną rzeczą jest m.in. znajomość różnych środowisk programistycznych i umiejętność wykorzystania ich funkcji. Czyni to wówczas ich pracę bardziej efektywną. Dowiemy się teraz, czym jest IDE – zintegrowane środowisko programowania, czyli podstawowe narzędzie pracy programisty.

Twoje cele

- Poznasz funkcje zintegrowanego środowiska programistycznego i dowiesz się, jak je wykorzystać do podniesienia wydajności pracy.
- Dowiesz się, jakie są najbardziej popularne środowiska programistyczne dla poszczególnych języków programowania.
- Przeanalizujesz, jakie zalety ma używanie środowiska programistycznego.

Przeczytaj

IDE – co warto wiedzieć?

IDE (*ang. Integrated Development Environment*) to zintegrowane środowisko programistyczne. Jest to program, w którym możemy tworzyć kod, testować go i zarządzać nim. IDE udostępnia w tym celu gotowe narzędzia.

Główni producenci IDE

W tym e-materiale przedstawiono największych producentów IDE oraz omówiono najpopularniejsze zintegrowane środowiska programistyczne.

- **Microsoft :**

- Microsoft Visual Studio – największe IDE firmy Microsoft. Możemy w nim programować m.in. w językach: C++, C#, Visual Basic, .NET, J Sharp. Pozwala tworzyć aplikacje dla systemów: Microsoft Windows, Windows 10 Mobile, XBOX i nie tylko (wersja Community na licencji darmowej).
- Visual Studio Code (TypeScript, JavaScript, CSS; licencja darmowa).
- Microsoft Access – aplikacja umożliwiająca zarządzanie relacyjnymi bazami danych. Wchodzi w skład pakietu Microsoft Office (VBA; licencja płatna).
- Microsoft Excel – zawiera w sobie IDE, Visual Basic Editor (VBE), służące do programowania w języku Visual Basic (licencja płatna).

- **Apple:**

- XCode – darmowe IDE udostępnione przez firmę Apple. Możemy je pobrać z AppStore. Służy do programowania aplikacji na wszystkie urządzenia Apple: macOS (MacBook), tvOS (AppleTV), iOS (iPhone) oraz watchOS (smartWatch). Posiada wiele różnorodnych funkcji, m.in. kontrolę wersji. Języki programowania, w których możemy pisać programy, to: C, C++, Swift oraz Objective-C.

- **Eclipse Foundation:**

- Eclipse IDE for Java Developers – jeden z najczęściej używanych IDE do programowania w języku Java. Umożliwia, dzięki Eclipse Marketplace, pobieranie i instalowanie własnych dodatków do IDE.
- Eclipse IDE for C/C++ Developers.
- Eclipse IDE for PHP Developers.
- Środowisko Eclipse można również skonfigurować do pracy z językiem **Python**.
- Licencja darmowa na wszystkie powyższe IDE.

- **JetBrains:**

- Posiada IDE do prawie każdego z popularniejszych języków programowania:
 - IntelliJ IDEA – język Java,
 - CLion – język C++,
 - PyCharm – język Python (w zależności od wersji: licencja płatna i darmowa),
 - PhpStorm – język PHP,
 - AppCode – język Swift, Objective-C (wersja Community – licencja darmowa, wersja Professional – płatna)
 - RubyMine – język Ruby.
- Licencja płatna na wszystkie powyższe IDE.
- **Oracle** – firma ta jest pionierem w tworzeniu IDE do zarządzania bazą danych.
 - **Oracle SQL Developer** – IDE służące do tworzenia, administrowania, zarządzania, projektowania relacyjnych baz danych. Umożliwia m.in. generowanie diagramów ERD (związków encji) oraz generowanie kodu SQL (licencja darmowa).
- **The Document Foundation:**
 - LibreOffice – program na licencji całkowicie darmowej. W jego skład wchodzi:
 - Writer (edytor tekstu),
 - Calc (arkusz kalkulacyjny),
 - Impress (prezentacje multimedialne),
 - Draw (wykresy i rysunki),
 - Base (baza danych),
 - Math (zapis równań matematyczne).
- **Bloodshed Software:**
 - Dev C++ – język C, C++ (licencja darmowa).
- **CodeBlocks:**
 - umożliwia pisanie w językach C, C++ (licencja darmowa).
- **FreePascal:**
 - umożliwia pisanie w języku Pascal (licencja darmowa).
- **IDLE:**
 - umożliwia pisanie w języku Python (licencja darmowa).

Korzyści z korzystania z IDE

Dzięki IDE możemy:

1. szybciej pisać programy;

- **Kolorowanie składni** – bardzo przydatna funkcjonalność zintegrowanych środowisk programistycznych. Dzięki niej kod jest o wiele bardziej czytelny. Pozwala na szybsze rozróżnianie elementów kodu – słów kluczowych, operatorów logicznych, zmiennych, tekstu, bibliotek. Każdy element składni ma inny kolor.

- **Podpowiadanie kodu** – zdecydowanie przyspiesza pisanie kodu oraz eliminuje błędy, które mogły powstać przez literówki. Podpowiadane są nie tylko słowa kluczowe danego języka, lecz także zadeklarowane już nazwy zmiennych, funkcje wbudowane itd.
- **Opisy używanych funkcji** – tę funkcjonalność ma np. Eclipse. Jeżeli będziemy chcieli użyć np. funkcji wbudowanej `length()` i napiszemy jej fragment, od razu pojawi się podpowiedź i okienko, które opisuje, jak działa ta funkcja, jakie argumenty przyjmuje i co jest wynikiem.
- **Możliwość dostosowania wyglądu** aplikacji wedle naszych preferencji – w większości IDE mamy dużą swobodę, jeżeli chodzi o zmianę tła, czcionki, kolorowania składni.

2. uruchamiać programy;

- **Automatyczne sprawdzanie poprawności kodu** – niezwykle przydatna funkcjonalność. Jeżeli popełnimy jakiś błąd w kodzie, nie tylko literówkę, lecz także błędy składniowe, IDE podkreśli to i wyświetli informację, jaki błąd został popełniony. Bardzo przydatna funkcja umożliwiająca eliminację większości błędów przed skompilowaniem i uruchomieniem programu.
- **Wbudowane narzędzia** – np. konsola, serwer lokalny. Dzięki nim oszczędzamy czas, a dodatkowo, jeżeli aplikacja, którą piszemy, wymaga serwera, nie musimy korzystać z zewnętrznych programów. Mamy to już udostępnione w tej samej aplikacji, w której piszemy kod aplikacji.

3. testować programy;

- **Debugger** – jedna z najczęściej używanych funkcjonalności IDE. Pozwala, krok po kroku, prześledzić działanie kodu. Widzimy wtedy, jak zmieniają się wartości zmiennych, możemy przejść przez kolejne iteracje. Jest to niezwykle przydatne, ponieważ nie musimy wypisywać zmiennych w kodzie, aby zobaczyć ich wartość. Jego główną ideą jest szybkie wyszukiwanie błędów w kodzie i ich eliminacja.

Słownik

funkcja wbudowana

napisana już funkcja w ramach konkretnego języka programowania, mająca określone zastosowanie; programista ma je do dyspozycji, aby efektywniej pisać kod; należy do jednej z bibliotek lub klas

IDE (Integrated Development Environment)

zintegrowane środowisko programistyczne; program, w którym możemy tworzyć kod, testować go i zarządzać nim; IDE udostępnia w tym celu gotowe narzędzia

kompilator

narzędzie tłumaczące cały kod programu na język maszynowy; następnie jest tworzony plik wykonywalny, który można uruchomić na innym komputerze bez użycia środowiska programistycznego

debugger

funkcjonalność służąca do szybkiego znajdowania błędów w kodzie programu i ich eliminowania

Prezentacja multimedialna

Polecenie 1

Przeanalizuj prezentację. Dzięki niej poznasz kilka funkcjonalności zintegrowanego środowiska programowania – DevC++.

Polecenie 2

Przeanalizuj prezentację. Dzięki niej poznasz kilka funkcjonalności zintegrowanego środowiska programowania – Eclipse IDE for Java Developers.

Polecenie 3

Przeanalizuj prezentację. Dzięki niej poznasz kilka funkcjonalności zintegrowanego środowiska programowania – PyCharm.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Ćwiczenie 2



Połącz w pary środowiska IDE wraz z językami w nich dostępnymi.

Pascal, C++, Swift, Objective-C, PHP, Python, Swift, Objective-C, Java

XCode	
PHPStorm	
AppCode	
FreePascal	
IDLE	
Eclipse	
CodeBlocks	

Ćwiczenie 3



Dokończ zdanie.

PyCharm to IDE firmy JetBrains przeznaczone do pisania programów w języku...

- C++.
- Java.
- Python.
- Swift.

Ćwiczenie 7



Wskaż, w jakich językach programowania można pisać kod w środowisku XCode.

- C++
- C#
- Swift
- Objective-C
- C

Ćwiczenie 8



Wskaż, które IDE nie pozwalają pisać programów w języku C++.

- XCode
- IntelliJ IDEA
- Eclipse IDE for Java Developers
- PyCharm
- DevC++

Dla nauczyciela

Autor: Maurycy Gast

Przedmiot: Informatyka

Temat: Zintegrowane środowisko programistyczne

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres rozszerzony

Podstawa programowa:

Cele kształcenia – wymagania ogólne

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania – wymagania szczegółowe

II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera i innych urządzeń cyfrowych.

Zakres rozszerzony. Uczeń spełnia wymagania określone dla zakresu podstawowego, a ponadto:

- 1) projektuje i tworzy rozbudowane programy w procesie rozwiązywania problemów, wykorzystuje w programach dobrane do algorytmów struktury danych, w tym struktury dynamiczne i korzysta z dostępnych bibliotek dla tych struktur;
- 2) stosuje zasady programowania strukturalnego i obiektowego w rozwiązywaniu problemów;
- 3) sprawnie posługuje się zintegrowanym środowiskiem programistycznym przy pisaniu, uruchamianiu i testowaniu programów;

Kształtowane kompetencje kluczowe:

- kompetencje cyfrowe;
- kompetencje osobiste, społeczne i w zakresie umiejętności uczenia się;
- kompetencje matematyczne oraz kompetencje w zakresie nauk przyrodniczych, technologii i inżynierii.

Cele operacyjne (językiem ucznia):

- Poznasz funkcje zintegrowanego środowiska programistycznego i dowiesz się, jak je wykorzystać do podniesienia wydajności pracy.
- Dowiesz się, jakie są najbardziej popularne środowiska programistyczne dla poszczególnych języków programowania.
- Przeanalizujesz, jakie zalety ma używanie środowiska programistycznego.

Strategie nauczania:

- konstruktywizm;
- konektywizm.

Metody i techniki nauczania:

- dyskusja;
- rozmowa nauczająca z wykorzystaniem multimediu i ćwiczeń interaktywnych;
- ćwiczenia praktyczne.

Formy pracy:

- praca indywidualna;
- praca w parach;
- praca w grupach;
- praca całego zespołu klasowego.

Środki dydaktyczne:

- komputery z głośnikami, słuchawkami i dostępem do internetu;
- zasoby multimedialne zawarte w e-materiale;
- tablica interaktywna/tablica, pisak/kreda;
- telefony z dostępem do internetu.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Zintegrowane środowisko programistyczne”. Uczniowie mają zapoznać się z treściami w sekcji „Przeczytaj”.
2. Chętny lub wybrany uczeń przygotowuje rozwiązanie polecenia nr 1 z sekcji „Prezentacja multimedialna”. Będzie pełnił rolę eksperta podczas zajęć.

Faza wstępna:

1. Prowadzący wyświetla na tablicy interaktywnej zawartość sekcji „Wprowadzenie” i omawia cele do osiągnięcia w trakcie lekcji.
2. **Rozpoznanie wiedzy uczniów.** Nauczyciel prosi wybranego ucznia lub uczniów o przedstawienie sytuacji problemowej związanej z tematem lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Uczniowie analizują treści z sekcji „Przeczytaj” wyświetlone na tablicy.
2. Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”, czyta treść polecenia nr 1 „Przeanalizuj prezentację” i omawia kolejne kroki rozwiązania. **Praca z multimedium.** Uczniowie w zespołach dwuosobowych zapoznają się z treścią polecenia nr 2 „Przeanalizuj prezentację” z sekcji „Prezentacja multimedialna” i wspólnie analizują kolejne kroki rozwiązania postawionego problemu.
3. **Ćwiczenie umiejętności.** Nauczyciel przechodzi do sekcji „Sprawdź się”. Uczniowie indywidualnie rozwiązują ćwiczenia nr 1-4 na czas. Osoba, która poprawnie rozwiąże zadania jako pierwsza, wygrywa, a nauczyciel może nagrodzić ją oceną za aktywność.
4. Nauczyciel dzieli uczniów na grupy czteroosobowe. Uczniowie wykonują ćwiczenia nr 5-8, a następnie porównują swoje odpowiedzi z inną grupą.

Faza podsumowująca:

1. Nauczyciel wyświetla na tablicy temat lekcji i cele zawarte w sekcji „Wprowadzenie”. W kontekście ich realizacji podsumowuje przebieg zajęć, a także wskazuje mocne i słabe strony pracy uczniów.
2. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności.

Praca domowa:

1. Uczniowie wykonują polecenie nr 3 „Przeanalizuj prezentację” z sekcji Prezentacja multimedialna.

Materiały pomocnicze:

- Oficjalna dokumentacja techniczna dla języka C++.
- Oficjalna dokumentacja techniczna dla kompilatora GCC/G++ 4.5 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.
- Oficjalna dokumentacja techniczna dla języka Java SE 8 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania Eclipse 4.4 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla języka Python 3 (lub nowszej wersji).
- Oficjalna dokumentacja techniczna dla oprogramowania PyCharm lub IDLE.

Wskazówki metodyczne:

- Treści w sekcji „Prezentacja multimedialna” można wykorzystać na lekcji jako podsumowanie i utrwalenie wiedzy uczniów.