



Wstęp do programowania w języku C++

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



Wstęp do programowania w języku C++

Źródło: Danielle MacInnes, domena publiczna.

Poznaliśmy już najpopularniejsze [języki programowania](#) oraz sposoby ich klasyfikacji. W e-materiale [Wstęp do programowania](#) poznaliśmy etapy powstawania programów komputerowych.

W tym e-materiale omówimy najpopularniejsze środowiska programistyczne C++, napiszemy także pierwszy program w tym języku.

Ciekawi cię, jak wyglądają te kwestie w innych językach programowania? Możesz się z nimi zapoznać w dwóch pozostałych lekcjach z tej serii:

- [Wstęp do programowania w języku Java](#),
- [Wstęp do programowania w języku Python](#).

Twoje cele

- Scharakteryzujesz najpopularniejsze środowiska programistyczne.
- Przećwiczysz, w jaki sposób dodawać komentarze i co zrobić, by kod był czytelny.
- Stworzysz i uruchomisz swój pierwszy program w języku C++.

Przeczytaj

Jak pisać przejrzysty kod?

W pracy programisty warto przestrzegać kilku zasad, aby napisany kod był czytelny i łatwy do analizowania. Należą do nich:

- używanie wcięć oraz umieszczanie każdej instrukcji w nowej linii:

```
1 #include <iostream>
2
3 int main() {
4     int i = 0;
5
6     while (i < 19) {
7         if (i == 7) {
8             std::cout << "Jestem siódmką" << std::endl;
9         }
10
11         std::cout << i << std::endl;
12         i++;
13     }
14
15     return 0;
16 }
```

- stosowanie odpowiednich nazw zmiennych,
- zapisywanie komentarzy.

Na początku przygody z programowaniem powinno się umieszczać w kodzie wiele komentarzy objaśniających wykorzystane instrukcje (przekonasz się niebawem, jak łatwo jest zapomnieć, do czego służą poszczególne sekwencje komend). Komentarze ułatwiają także analizowanie zapisanego algorytmu innym osobom. Pamiętaj, że wyjaśnienia umieszczone w kodzie są ignorowane przez kompilator i nie zwiększają rozmiaru pliku wykonywalnego.

Oto sposoby zamieszczania komentarzy w kodzie programu:

- **komentarz jednowierszowy**: po wpisaniu dwóch znaków slash (//) pozostała część wiersza jest traktowana jako komentarz:

```
1 int a = b + c; // suma dwóch liczb
```

- **komentarz wielowierszowy**: rozpoczyna się sekwencją `/*`, natomiast kończy znakami `*/` i może zająć więcej niż jedną linię:

```
1 int a = b + c; /* suma dwóch liczb,  
2 ten komentarz zajmuje dwie linie */
```

Środowisko programistyczne

Jak już wiemy z e-materiału [Wstęp do programowania](#) środowisko programistyczne poza pisanie kodu realizuje wiele funkcji, m.in. wykonuje kompilację programu, wskazuje niewłaściwie zapisaną instrukcję, podpowiada słowa kluczowe, udostępnia narzędzia pomocne podczas tworzenia interfejsu graficznego [interfejsu graficznego](#). Dlatego właśnie powinniśmy dobrać do własnych potrzeb odpowiednie środowisko ułatwiające pracę z kodem.

Słownik

graficzny interfejs użytkownika (GUI)

sposób prezentacji danych i komunikacji z użytkownikiem za pomocą takich elementów, jak okna, suwaki, przyciski itp.

liczby binarne

liczby, które są zapisywane przy użyciu dwóch cyfr: 0 oraz 1; podstawą systemu binarnego jest liczba 2; systemem takim posługują się m.in. procesory

Prezentacja multimedialna

Polecenie 1

Napisz program, który wyświetli napis *Cogito ergo sum*.

Twoje zadania

1. Program wypisuje napis *Cogito ergo sum*.

```
1 #include <iostream>
2
3 int main () {
4
5     // Tutaj dodaj kod. Żeby coś wypisać, użyj polecenia:
6     std::cout
7 }
```

```
1
```

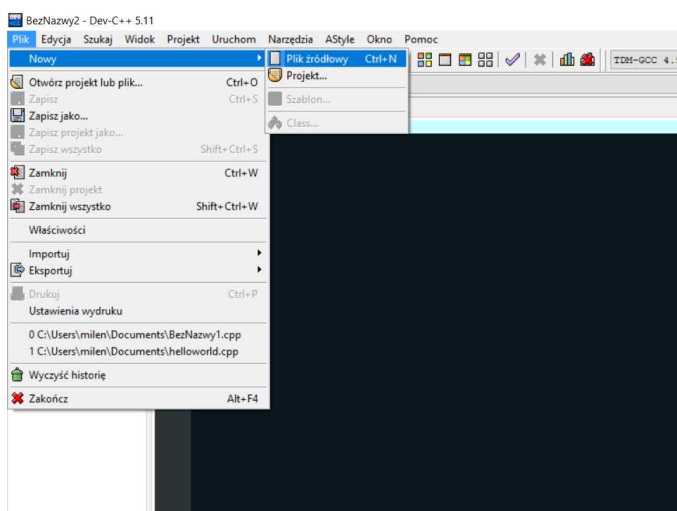
Polecenie 2

Zapoznaj się z prezentacją i porównaj z nią swoje rozwiązanie.

Prezentację zaczniemy od napisania programu w środowisku DEV-C++ oraz Code::Blocks. Pobierzmy je.

1

2

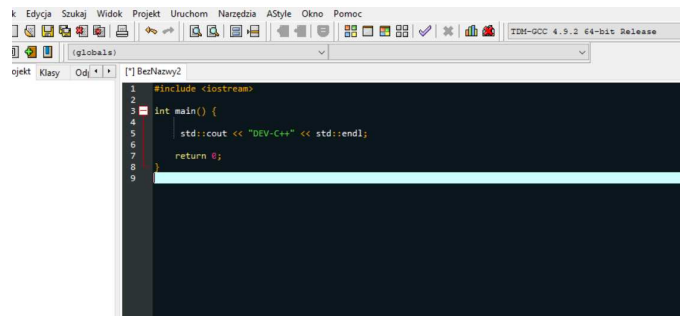


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Rozpocznijmy od utworzenia nowego pliku. Wydajemy polecenie Plik | Nowy | Plik źródłowy. Możemy także użyć kombinacji klawiaturowej `Ctrl+N`.



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

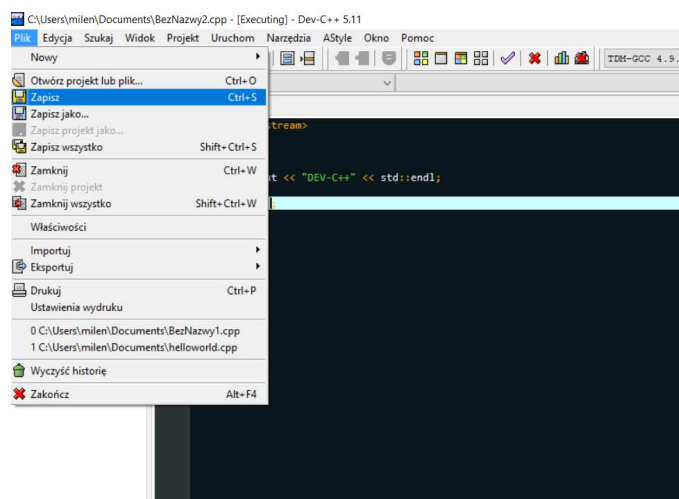
<https://zpe.gov.pl/b/PawRAntoa>

W pliku źródłowym umieścimy kod programu wyświetlającego na ekranie tekst: DEV-C++:

```

1 #include <iostream>
2
3 int main() {
4     std::cout << "DEV-C++"
5     << std::endl;
6
7     return 0 ;
8 }

```

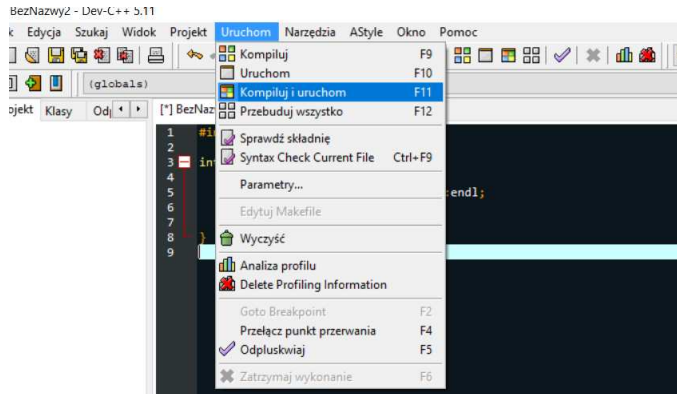


Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Zapiszmy nasz plik. W tym celu wydajemy polecenie Plik | Zapisz. Pamiętajmy, by użyć rozszerzenia .cpp.



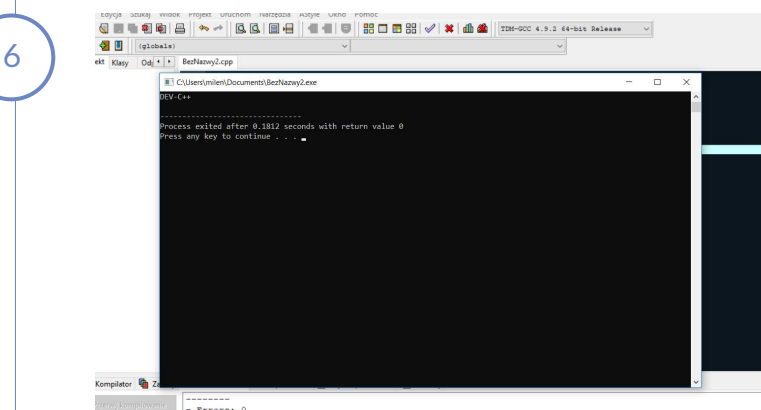
5

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Aby skompilować i uruchomić program, klikamy Uruchom, a następnie Kompiluj i uruchom. Można także użyć klawisza F11.



6

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Na ekranie pojawił się napis: DEV-C++. Program działa więc poprawnie.

7

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Środowisko DEV-C++ jest polecane głównie początkującym programistom. Do zaawansowanych zastosowań warto wybrać jednak bardziej rozbudowany pakiet.

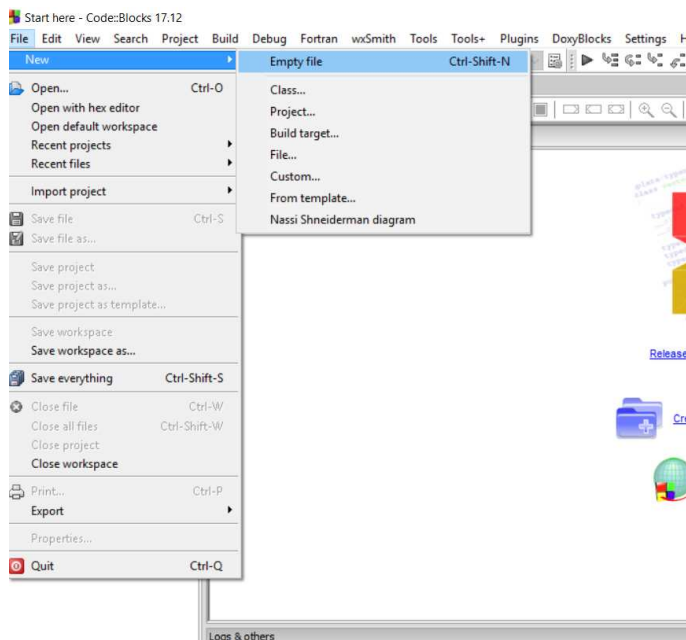
8

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Napiszemy program podobny do przedstawionego przed chwilą. Tym razem użyjemy środowiska Code::Blocks.

9



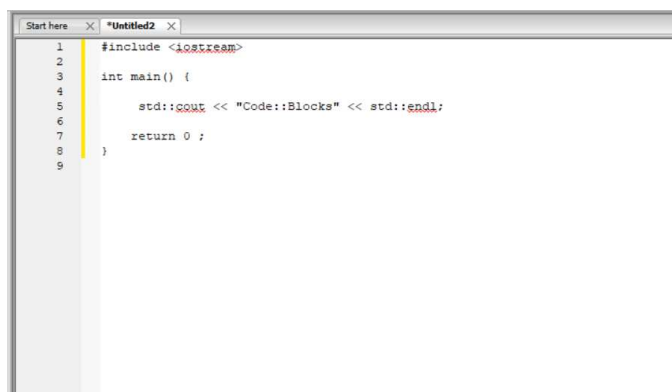
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Utworzymy nowy plik źródłowy. W tym celu wydajemy polecenie File | Empty file. Możemy także użyć kombinacji klawiaturowej Ctrl+Shift+N.

10



```
1 #include <iostream>
2
3 int main() {
4
5     std::cout << "Code::Blocks" << std::endl;
6
7     return 0 ;
8 }
9
```

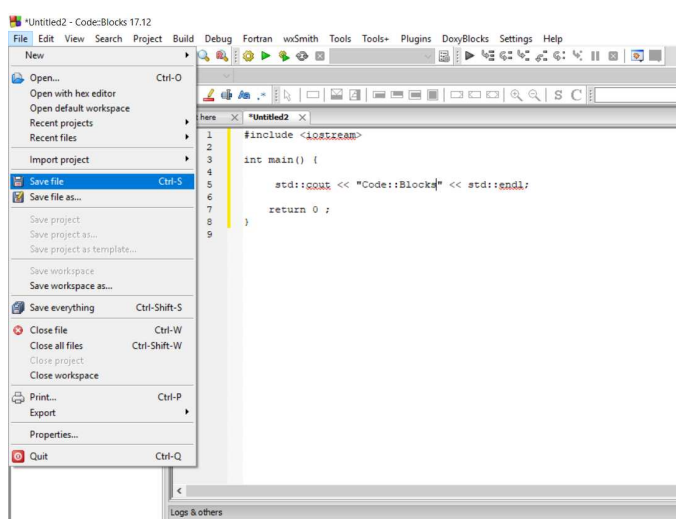
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

W pliku źródłowym zapisujemy kod podobny do poprzedniego, jednak teraz chcemy, by program wyświetlił napis Code::Blocks.

11



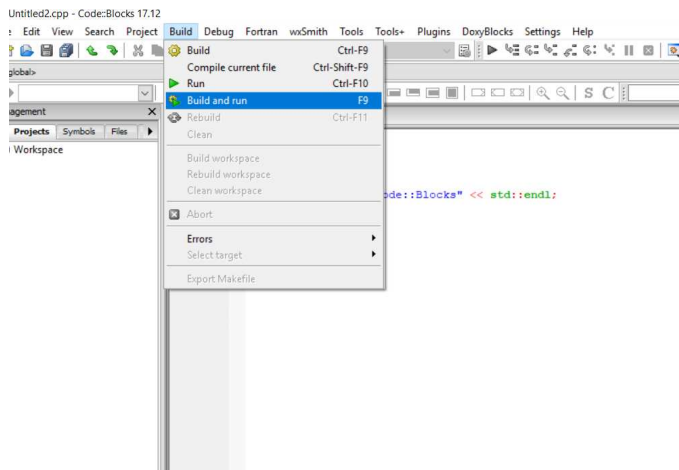
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Zapisujemy zmiany w pliku. Możemy wykorzystać skrót klawiaturowy `Ctrl+S`. Pamiętajmy o rozszerzeniu `.cpp`.

12



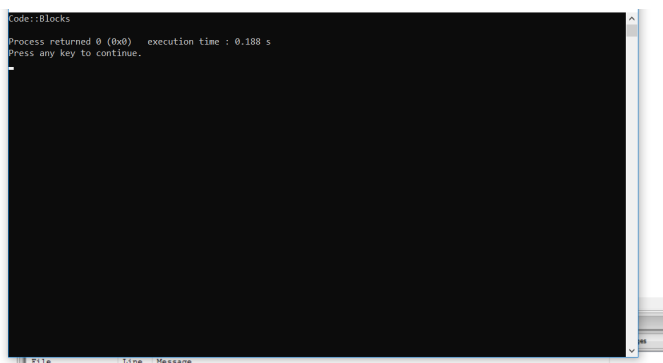
Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Kompilujemy i uruchamiamy program, wydając komendę `Build` | `Build and run` lub naciskając klawisz `F9`.

13



Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Na ekranie pojawia się napis `Code::Blocks`.

Materiał audio dostępny pod adresem:

<https://zpe.gov.pl/b/PawRAntoa>

Code::Blocks działa na platformach Windows, MacOS X i Linux. Pomimo tego, że pakiet jest darmowy, obsługuje wiele opcji przydatnych nawet zaawansowanym programistom.

Źródło: Contentplus.pl Sp. z o.o., licencja: CC BY-SA 3.0.

Polecenie 3

Zapisz notatkę, w której podsumujesz najważniejsze informacje przedstawione w prezentacji multimedialnej.

Sprawdź się

Pokaż ćwiczenia:   

Ćwiczenie 1



Wskaż, co poprawia czytelność kodu.

- ☐ Używanie komentarzy.
- ☐ Kompilowanie programu po każdej napisanej instrukcji.
- ☐ Używanie wcięć.
- ☐ Umieszczanie każdej instrukcji w nowej linii.

Ćwiczenie 2



Zdecyduj, które zdania dotyczące komentarzy zamieszczanych w kodzie źródłowym są prawdziwe, a które fałszywe.

Stwierdzenie	Prawda	Fałsz
Sprawiają, że kod jest bardziej czytelny.	☐	☐
Sprawiają, że osoba, która pracuje z kodem będzie wiedziała, do czego służy dana instrukcja.	☐	☐
Przyspieszają kompilację programu.	☐	☐

Ćwiczenie 3



Wyjaśnij, w jaki sposób oznaczamy komentarze w języku C++.

Ćwiczenie 4



Wskaż funkcje, jakie może realizować środowisko programistyczne.

☐ kompilacja programu

☐ zaznaczanie błędów

☐ edycja kodu

Ćwiczenie 5



Wskaż język, którym posługuje się procesor.

☐ C++

☐ ciąg instrukcji zapisanych w postaci liczb dziesiętnych

☐ każdy język programowania

☐ ciąg instrukcji zapisanych w postaci liczb binarnych

Ćwiczenie 6



Wskaż, czym zajmuje się translator.

☐ poprawieniem błędów w kodzie

☐ tłumaczeniem kodu napisanego w dowolnym języku programowania na język maszynowy

☐ uruchamianiem programu

Ćwiczenie 7



Wybierz środowisko przeznaczone dla bardziej zaawansowanych użytkowników.

☐ Code::Blocks

☐ Dev-C++

Ćwiczenie 8



Zdecyduj, które zdania dotyczące kompilatora są prawdziwe.

- ☐ Tłumaczy cały napisany kod źródłowy.
- ☐ Nie możemy go uruchomić bez środowiska programistycznego.
- ☐ Tłumaczy kod źródłowy metodą instrukcja po instrukcji.
- ☐ Tworzy plik wykonywalny .exe.

Ćwiczenie 9



Popraw kod.

Twoje zadania

1. Program wypisuje napis Hello World!

```
1 #include <iostream>
2
3 int main() {
4     std::cout << Hello World!;
5     return 0;
6 }
```

1



Dla nauczyciela

Autor: Zespół autorski [Contentplus.pl](https://contentplus.pl) sp. z o.o.

Przedmiot: Informatyka

Temat: Wstęp do programowania w języku C++

Grupa docelowa:

Liceum ogólnokształcące i technikum, liceum ogólnokształcące, technikum, zakres podstawowy

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 podstawowy. Uczeń:

2) do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje oraz zasoby, wykorzystuje również elementy robotyki;

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):

- Scharakteryzujesz najpopularniejsze środowiska programistyczne.
- Przećwiczysz, w jaki sposób dodawać komentarze i co zrobić, by kod był czytelny.
- Stworzysz i uruchomisz swój pierwszy program w języku C++.

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;
- oprogramowanie dla języka C++, w tym kompilator GCC/G++ 4.5 (lub nowszej wersji) i Code::Blocks 16.01 (lub nowszej wersji), Orwell Dev-C++ 5.11 (lub nowszej wersji) lub Microsoft Visual Studio.

Przebieg lekcji

Przed lekcją:

1. **Przygotowanie do zajęć.** Nauczyciel loguje się na platformie i udostępnia e-materiał: „Wstęp do programowania w języku C++”. Uczniowie zapoznają się z treściami w sekcji „Przeczytaj”.

Faza wstępna:

1. Wyświetlenie przez nauczyciela tematu i celów lekcji. Określenie wiążących dla uczniów kryteriów sukcesu.
2. **Rozpoznanie wiedzy uczniów.** Uczniowie tworzą pytania dotyczące tematu zajęć, na które odpowiedzą w trakcie lekcji.

Faza realizacyjna:

1. **Praca z tekstem.** Nauczyciel ocenia, na podstawie informacji na platformie, stan przygotowania uczniów do zajęć. Jeżeli jest ono niewystarczające prosi o ciche zapoznanie się z treścią w sekcji „Przeczytaj”.

2. **Praca z multimediami.** Nauczyciel wyświetla zawartość sekcji „Prezentacja multimedialna”. Uczniowie wspólnie zapoznają się z prezentacją, z której dowiedzą się, jak napisać program, korzystając z dwóch darmowych środowisk programistycznych.
3. **Ćwiczenie umiejętności.** Uczniowie wykonują indywidualnie ćwiczenia nr 1-8, a następnie dzielą się wynikami swojej pracy z kolegą lub koleżanką.

Faza podsumowująca:

1. Wybrany uczeń podsumowuje zajęcia, zwracając uwagę na nabyte umiejętności, omawia ewentualne problemy podczas rozwiązywania ćwiczeń z programowania w języku C++.

Praca domowa:

1. Uczniowie wykonują program przedstawiony na filmie 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.

Wskazówki metodyczne:

- Treści w sekcji „Przeczytaj” można wykorzystać jako podsumowanie i utrwalenie wiedzy uczniów.