

PRIMM: NOWY SPOSÓB NA STARE PROBLEMY Z NAUKĄ PROGRAMOWANIA

Agnieszka Borowiecka, Katarzyna Olędzka
Ośrodek Edukacji Informatycznej i Zastosowań Komputerów
w Warszawie, Raszyńska 8/10
{agnieszka.borowiecka, katarzyna.oledzka}@oeiizk.edu.pl

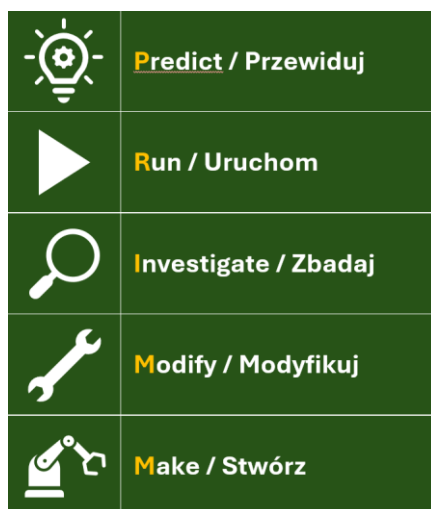
Abstract. *Teaching programming requires a well-planned educational process. The PRIMM method can be a remedy for the commonly known problems associated with learning programming. This method is based on several steps: students predict the program's operation, then run it, and then develop it. Thanks to this, they learn to consciously analyze, modify, and create programs – in block and text languages.*

1. Wstęp

Nauczanie programowania w szkole nie jest proste. Często w klasie mamy uczniów o różnym poziomie zaawansowania i zaangażowania, mających problemy z logicznym myśleniem lub braki w wiedzy matematycznej. Samodzielne ułożenie i zapisanie algorytmu sprawiają im trudność, bowiem nie wiedzą, od czego powinni zacząć. Podanie przez nauczyciela gotowego rozwiązania nie prowadzi do rozwoju myślenia algorytmicznego. Przy doborze zadań i sposobu organizacji lekcji warto skorzystać z metody PRIMM, która w praktyczny sposób prowadzi ucznia krok po kroku przez proces nauki programowania. W ten sposób lekcje stają się bardziej przystępne i angażujące uczniów.

2. Poznajemy model PRIMM

Model PRIMM to podejście dydaktyczne wyróżniające pięć kluczowych etapów nauki: **Predict** (Przewiduj), **Run** (Uruchom), **Investigate** (Zbadaj), **Modify** (Modyfikuj) oraz **Make** (Stwórz). Dzięki tej metodzie nauczyciel otrzymuje strukturę porządkującą proces nauczania, a uczniowie mogą krok po kroku opanować sztukę rozwiązywania zadań algorytmicznych i rozwijać myślenie komputacyjne.

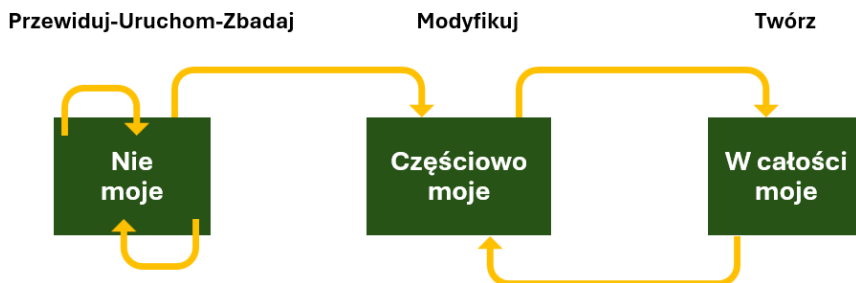


Rysunek 1. Infografika przedstawiająca metodę PRIMM

Planując lekcję lub cykl lekcji z wykorzystaniem metody PRIMM zakładamy następujące etapy pracy:

- Uczniowie otrzymują gotowy kod programu, nie uruchamiają go tylko analizują i przewidują (**Predict**), jakie będzie jego działanie. Dzięki temu rozwijają umiejętność logicznego myślenia.
- Uczniowie uruchamiają program (**Run**), by sprawdzić swoje przewidywania i zweryfikować, czy działanie programu jest zgodne z tym, czego oczekiwali.
- Kolejny krok polega na badaniu struktury i szczegółów kodu (**Investigate**). Dzięki temu uczniowie mogą lepiej zrozumieć zastosowane rozwiązania programistyczne.
- Uczniowie wprowadzają zaproponowane modyfikacje (**Modify**), eksperymentują i uczą się rozwiązywania problemów.
- Ostatnim etapem jest tworzenie własnego programu (**Make**), który pozwala kreatywnie wykorzystać zdobyte umiejętności.

Warto zauważyć, że nie wszystkie wymienione fazy muszą wystąpić na każdej lekcji. Jednocześnie pierwsze trzy lub cztery etapy można powtarzać kilka razy, kładąc za każdym razem nacisk na trochę inny aspekt prezentowanych treści lub wykonywanych zadań.



Rysunek 2. Diagram ilustrujący działanie metody PRIMM

Model PRIMM został zaproponowany przez Sue Sentance i Jane Waite z King's College London [5]. Projekt rozpoczął się w 2017 roku od 8-tygodniowego szkolenia, dotyczącego nauczania informatyki (ang. *computer science*) w klasach 6-8, dla 15 nauczycieli bez wcześniejszego doświadczenia. W drugim etapie zaproponowano siedmiu doświadczonym nauczycielom zastosowanie podejścia PRIMM podczas prowadzonych przez nich lekcji przez okres 4-6 tygodni. Zarówno przed rozpoczęciem, jak po zakończeniu zajęć uczniowie oraz nauczyciele zostali poproszeni o wypełnienie ankiety wielokrotnego wyboru.

Do Now in pairs

Here is some code written in Python

```

name = "Billy"
print("We want to know if you like programming!")
print()
print("Do you like programming " + name + "?")
answer = input()
print("Great! You said " + answer + "!")
print("Let's learn some Python today")

```

Write out *exactly* what you think will happen when you run the Python code

Rysunek 3. Karta pracy dla ucznia podchodząca z portalu PRIMM [6]

W 2017 roku powstało kilka arkuszy roboczych, które zostały użyte w badaniu pilotażowym [6]. W 2018 roku przeprowadzono kolejne badanie z grupą szkół, podczas którego wykorzystano nowe materiały przeznaczone dla uczniów już mających niewielkie doświadczenie w programowaniu. Każdy z przygotowanych zestawów

(10 lekcji/tygodni zajęć) obejmuje materiały wprowadzające, plany lekcji, aktywności itp. Materiały te (w języku angielskim) są dostępne w portalu PRIMM [6], skąd można je pobrać i dostosować do własnych potrzeb.

3. Przykładowe materiały z portalu PRIMM

Przyjrzyjmy się bliżej propozycji lekcji dotyczącej wprowadzenia instrukcji warunkowej oraz definiowania funkcji. Uczniowie otrzymują kartę pracy z fragmentem programu w języku Python i poleceniem przedyskutowania w parach, co ich zdaniem wydarzy się po jego uruchomieniu. Swoje przemyślenia mają zapisać na kartce.

Kolejny dokument to szczegółowy plan pracy dla ucznia podzielony na kolejne zadania, z miejscem na notowanie uwag i spostrzeżeń. Zadanie 3 (Task 3) dotyczy modyfikowania otrzymanego programu. Uczniowie mogą przy okazji nauczyć się, w jaki sposób środowisko, w którym pracują, sygnalizuje wystąpienie błędu w kodzie, np. błędną pisownię polecenia `print()`.

```
def conversation():
    print("Welcome to my conversation program")
    print()
    print("Do you like cycling? Answer yes or no")
    answer = input()
    if answer == "yes":
        print("That's good - you will get very fit")
    else:
        print("Perhaps you like some other sport. ")
    print("Goodbye")

conversation()
```

What would you expect the computer to do? Write it out exactly.

Task 2: Download and run the program and see if it does what you think it might do. You will find it at <insert your shared drive here>

Did the program run as you predicted? _____

Task 3: There are a few new concepts in this program. In pairs, work out the answers to the following questions by examining the code and running it a few times.

1. What would happen if the program used the word `Print()` instead of `print()`?

.....

Rysunek 4. Fragment planu pracy ucznia

Nauczyciel może skorzystać z pliku z prezentacją przedstawiającą kolejne etapy lekcji, w tym omawianą tematykę, poszczególne zadania do wykonania wraz z czasem, jaki należy na nie poświęcić, oraz pytania na jakie mają odpowiedzieć uczniowie. Dostępne są również dwa pliki z wersją startową omawianego programu oraz przykładowymi propozycjami modyfikacji/rozbudowy programu na bazie uzyskanej wiedzy.

```
Sue Sentence udostępnił(a) ten plik. Chcesz coś z nim zrobić? Zaloguj się Zarejestruj się

1 # starter program week 2 - perhaps this could be a task
2
3 def conversation():
4     print("Welcome to my conversation program")
5     print()
6     print("Do you like cycling? Answer yes or no")
7     answer = input()
8     if answer == "yes":
9         print("That's good - you will get very fit")
10    else:
11        print("Perhaps you like some other sport. ")
12    print("Goodbye")
13
14 # conversation()
15
16 ##1.    Modify the program to have a conversation with somebody #
17 ## about sport. An example is given below but you can add your own questions.
18
19 def conversation_sport():
20     print("Welcome to my sport discussion program")
21     print()
22     print("Do you like running")
23     answer = input()
24     if answer == "yes":
25         print("That's good - which runner do you admire")
26         answer = input()
27         if answer == "Mo Farah":
28             print("Yes he is a great runner ")
29         else:
30             print("That's interesting")
31     else:
32         print("Do you watch the London marathon? ")
33         answer = input()
34         if answer == "yes":
35             print("That's good - it is inspiring")
36     print("Goodbye")
```

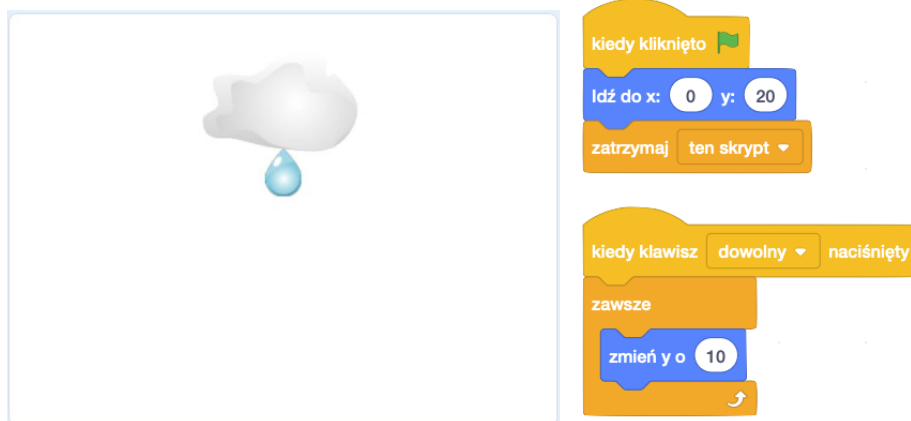
Rysunek 5. Fragment udostępnionego programu w języku Python

W sieci znajdziemy również inne materiały dotyczące nauki programowania z wykorzystaniem metody PRIMM przeznaczone dla różnych poziomów edukacyjnych. Zachęcamy do eksploracji zasobów dostępnych na stronie National Centre for Computing Education [4].

4. PRIMM na naszej lekcji

Podejście PRIMM możemy zastosować podczas lekcji dotyczących programowania w środowisku Scratch. Zadaniem uczniów będzie przygotowanie animacji deszczu padającego z chmury. Przygotowujemy kartę pracy, na której widoczne jest początkowe położenie duszków „chmura” i „kropla” oraz skrypty dla duszka „kropla”. Zachęcamy uczniów do zastanowienia się, jaki będzie efekt działania programu: co stanie się po

naciśnięciu zielonej flagi, co robi duszek „kropla” po naciśnięciu klawisza na klawiaturze, jaki klawisz powinni nacisnąć, by uruchomić skrypt, itp.



Rysunek 6. Wersja wstępna aplikacji wraz ze skryptami duszka „kropli”

Następnie uczniowie otrzymują link do programu i uruchamiają go, sprawdzając, czy kropla porusza się zgodnie z ich oczekiwaniami. W kolejnym kroku dokonujemy prostych modyfikacji, np.:

- zmieniamy kierunek ruchu kropli,
- zmieniamy klawisz uruchamiający skrypt,
- zmieniamy położenie początkowe kropli (ukryta za chmurą),
- dodajemy bloczek *czekaj* w celu spowolnienia ruchu,
- badamy, jakie wartości należy użyć w bloczku *czekaj*, by ruch odbywał się bardziej płynnie.

Kolejne modyfikacje projektu mogą wymagać dodatkowej wiedzy. Uczniowie mogą zauważyć, że kropla zatrzymuje się przy dolnej krawędzi sceny i nie zmienia swojego położenia mimo ciągle wykonywanej pętli *zawsze*. Możemy wrócić do etapu przewidywania (Predict) i udostępnić kolejny fragment skryptu.



Rysunek 7. Pytanie: Jaki jest wynik działania tego fragmentu skryptu?

Tym razem działanie uczniów polega na dodaniu odpowiednich blozków do skryptu opisującego ruch duszka „kropli” i przetestowanie działania. Pierwsza modyfikacja dotyczy miejsca, w którym znika kropka. Uczniowie powinni zauważyć pojawiający się problem – po ukryciu duszka musimy zadbać o jego ponowne pojawienie się. Przykładowe kolejne zmiany mogą polegać na:

- dodaniu wielu duszków „kropli” do projektu,
- zatrzymaniu skryptu w momencie, gdy kropka dotrze do krawędzi sceny,
- przemieszczeniu duszków do położenia początkowego za chmurą, po dotarciu przez nie do krawędzi sceny,
- wprowadzeniu odstępu czasu, zanim kolejna kropka zacznie spadać,
- zmianie wielkości i koloru kropki.

Uczniowie mogą teraz przygotować dowolny własny projekt wykorzystujący animację duszków, np. do przedstawienia ruchu samolotu po niebie, jabłek spadających z drzewa, czy startującej rakiety.

Taki sam styl pracy można wykorzystać także w pracy ze starszymi uczniami przy bardziej zaawansowanych projektach, na przykład do zaprogramowania gry *Wisielec*. Jest to ogólnie znana gra pozwalająca wykorzystać różne konstrukcje programistyczne i wykazać się kreatywnością. Uczniowie zaczną od analizy gotowego kodu, następnie rozwijając stopniowo umiejętności programistyczne przygotowują własną wersję gry.

```
1. slowo = "informatyka"
2. odgadnione = ["_"] * len(slowo)
3. proby = 5
4.
5. while proby > 0 and "_" in odgadnione:
6.     print(" ".join(odgadnione))
7.     print("Zgadnij literę, zostało prób", proby)
8.     litera = input()
9.
10.    if litera in slowo:
11.        for i in range(len(slowo)):
12.            if slowo[i] == litera:
13.                odgadnione[i] = litera
14.    else:
15.        proby -= 1
16. if "_" not in odgadnione:
17.     print("Wygrałeś!")
18. else:
19.     print("Przegrałeś!")
```

Rysunek 8. Implementacja gry Wisielec w języku Python

Przed uruchomieniem programu uczniowie mogą przewidzieć (**Predict**), co się stanie: jak wyświetlany jest aktualny stan odgadywanego słowa, ile razy użytkownik podaje odgadywaną literę, jak sprawdzana jest poprawność odpowiedzi, co się dzieje, gdy popełnimy błąd, kiedy zakończy się działanie programu. Podczas testowania działania programu (**Run**) należy podkreślić konieczność wykonania wielu prób oraz sprawdzenia działania programu dla poprawnych i niepoprawnych danych.

Analizując poszczególne elementy programu (**Investigate**) uczniowie mogą zwrócić uwagę na użycie zmiennych i ich wartości początkowe, konstrukcję warunków zakończenia pętli `while`, algorytm obsługi procesu zgadywania (sprawdzam czy litera jest w słowie, przeglądam słowo po indeksach i w odpowiednich miejscach zastępuję znak podkreślenia literą), sposób zakończenia gry (warunki wygranej i przegranej).

Uczniowie mogą następnie wprowadzić zmiany według własnego pomysłu: dodać więcej słów do odgadywania, zmodyfikować komunikaty, wprowadzić podpowiedzi (np. po dwóch błędnych próbach) lub nowe funkcjonalności (np. opcja odgadnięcia całego słowa od razu, zmiana liczby prób w zależności od długości słowa). Można też dodać ilustrację graficzną za pomocą kombinacji znaków ASCII lub grafiki żółwia (moduł `turtle`).

5. Z perspektywy nauczyciela

Zastosowanie etapów pracy (**Predict – Run – Investigate – Modify – Make**) porządkuje cały proces, pozwala stopniowo poszerzać wiedzę i rozwijać umiejętności. Łatwiej jest dobrać poziom trudności zadań, stymulować do kreatywnego myślenia, ale także uczyć analizowania rozwiązań innych osób. Uczniowie najpierw próbują zrozumieć gotowy kod, a dopiero potem modyfikować go i tworzyć własne rozwiązania, co rozwija ich umiejętność analizy i krytycznego myślenia. PRIMM sprawdza się szczególnie dobrze w klasach zróżnicowanych pod względem umiejętności, gdyż pozwala każdemu uczniowi pracować we własnym tempie.

W metodzie PRIMM nauczyciel pełni rolę przewodnika, wspierając uczniów, kierując ich uwagę na kluczowe aspekty pracy i wyjaśniając trudniejsze elementy. Zadając pytania skłaniające do refleksji pomagamy uczniom zrozumieć działanie programu, sugerując nowe możliwości lub podpowiadając kierunki rozwiązań wspieramy ich kreatywność. Ważne jest umiejętne zachęcanie do samodzielnego myślenia i eksperymentowania. Musimy przy tym zadbać o to, by każdy uczeń mógł odnieść sukces na swoim poziomie zaawansowania.

6. Podsumowanie

Zapewne nie wszystkie lekcje będziemy prowadzić wykorzystując metodę PRIMM, jednak warto rozważyć jej stosowanie przynajmniej na części zajęć. Możemy korzystać z gotowych materiałów i pomysłów dostępnych w internecie, lub zaproponować własne

rozwiązania. Metoda PRIMM to coś więcej niż tylko narzędzie dydaktyczne – to sposób na uczynienie nauki programowania bardziej dostępnej, zrozumiałej i angażującej dla uczniów. Pozwala ona każdemu, niezależnie od poziomu zaawansowania, krok po kroku budować własne kompetencje i odczuwać radość z tworzenia programów.

Literatura

1. Brasier O., Teaching Coding and Digital Technologies Using Scratch, Medium 2023, <https://tiny.pl/ck365rtt>, ostatni dostęp 5.06.2025 roku.
2. National Centre for Computing Education, *Quick Read: Using PRIMM to structure programming lessons*, <https://tiny.pl/jfzzf37k>, ostatni dostęp 4.06.2025 roku.
3. National Centre for Computing Education, *Teach Computing Curriculum*, <https://teachcomputing.org/curriculum>, ostatni dostęp 4.06.2025 roku.
4. Olędzka K., *Metoda PRIMM – nauka programowania krok po kroku*, kwartalnik W cyfrowej szkole 1/2025, OEliZK, Warszawa 2025.
5. PRIMM Portal, <https://primmportal.com>, ostatni dostęp 4.06.2025 roku.
6. Sentance S., J. Waite, *PRIMM: Exploring pedagogical approaches for teaching text-based programming in school*, King's College London 2017, https://tiny.pl/1m_xn8qm, ostatni dostęp 4.06.2025 roku.