

## **TRUDNE TEMATY W NAJPROSTSZY SPOSÓB: WPROWADZENIE DO ZMIENNYCH**

*Damian Kurpiewski*  
*Instytut Podstaw Informatyki, PAN*  
*d.kurpiewski@ipipan.waw.pl*  
*Krzysztof Skowronek*  
*Wydział Matematyki i Informatyki UMK*  
*krzysko@gmail.com*

*Abstract. It is a well-known fact, that learning computer science, especially programming, is not an easy task. One of the topics that can cause many problems, when learning the art of the programming, is the topic connected with the variables. In this paper we present a simple exercise, that can help students understand complex programming procedures, including, but not limited to, operations on the variables.*

### **1. Wstęp**

Przygoda z programowaniem zaczyna się w coraz młodszym wieku. Już w przedszkolu dzieci mają nieraz szansę na stworzenie własnej gry, wykorzystując do tego proste narzędzia do programowania wizualnego. W szkole podstawowej nauka algorytmiki i programowania jest już wpisana w podstawę programową [2,3]. Im wcześniej zagłębiamy się w tajemniczy świat pisania własnych gier i programów, tym ważniejsze jest byśmy dobrze poznali i zrozumieli jego podstawy, aby później się nie pogubić i nie zniechęcić. Programowanie jest bardzo przydatną i cenioną umiejętnością, nie mniej zdobycie jej wiąże się z dużym trudem, wysiłkiem i latami ciężkiej pracy.

Nauczając w szkole programowania, szczególnie jego początkowych etapów, łatwo można zauważyć, czy uczniowie dobrze przyswoili pewne podstawy. Uczeń, który nie rozumiał w wystarczającym stopniu pewnych podstawowych zasad, popełnia dość typowe błędy. Jednym z takich tematów, które należy dobrze przyswoić, aby móc tworzyć różnorodne programy, jest temat zmiennych. Użycie zmiennych w programowaniu jest niezbędne, aby napisać chociaż trochę bardziej zaawansowany program. To za pomocą zmiennych odwołujemy się do pamięci komputera – zapisujemy w niej pewne dane, by później móc je odczytać, zmodyfikować i przeprowadzić na nich najróżniejsze operacje. Ze zmiennymi wiąże się kilka zagadnień, wśród których za naj-

ważniejsze uznać można rozmiary i typy danych, a także sposób ich przechowywania w pamięci komputera. Aby móc korzystać ze zmiennych, nie jest niezbędnym wiedzieć, jak dokładnie komputer przedstawia i przechowuje je w swojej pamięci, jednak pewna intuicja może okazać się bardzo pomocna. Tworząc program powinniśmy zawsze mieć z tyłu głowy świadomość, że pracujemy z maszyną, która może wykonać jedynie to, do czego została zaprogramowana.

Struktura tego artykułu przedstawia się następująco. Zaczniemy od zaprezentowania możliwych trudności, jakie uczeń może napotkać podczas próby zrozumienia działania zmiennych w programowaniu, a także wyjaśnimy, dlaczego proste i wręcz łopatologiczne przedstawienie tego tematu jest takie ważne. W kolejnym rozdziale opiszemy strukturę stworzonego tematu i ćwiczeń. Na koniec opowiemy o dalszych planach i wyciągniemy wnioski z dotychczasowych doświadczeń.

## 2. Zmienne – trudność tematu

Zmienne nie uchodzą powszechnie za temat trudny. Oczywiście, jak niemalże z każdym innym tematem związanym z programowaniem, zdarza się, iż także ten sprawia uczniom niemałe problemy. Zazwyczaj jednak, gdy jest przeprowadzony na odpowiednim etapie edukacyjnym, nie wydaje się sprawiać uczniom większych problemów. Także sami uczniowie są zazwyczaj przekonani, że temat zrozumieli. Samo pojęcie zmiennej w programowaniu, a także jej wykorzystanie, abstrahując już od różnych typów zmiennych i danych, nie wydaje się być trudne do przyswojenia. W tym właśnie tkwi haczyk – temat **wydaje się** być stosunkowo prosty. I rzeczywiście, w trakcie poznawania zmiennych i wykonywania prostych ćwiczeń z ich wykorzystaniem, uczniowie rzadko mają problemy - te pojawiają się później.

Skąd więc bierze się trudność tematu zmiennych? Ma ona kilka źródeł. Nie można ukryć, że tematowi poświęca się zazwyczaj zbyt mało czasu i uwagi. Dzieje się tak dlatego, że uchodzi on za stosunkowo prosty, a także mało obszerny. Poza tym zmienne są wykorzystywane przy tworzeniu niemalże każdego programu, uczniowie będą więc je ćwiczyć praktycznie na każdej lekcji.

Druga trudność wynika z braku odpowiedniej koordynacji pomiędzy tematami na lekcjach informatyki i matematyki, a także na powielaniu nazewnictwa. Należy pamiętać, że zmienna w programowaniu, a zmienna w matematyce, to dwa różne twory. Na etapie szkolnym zmienne w matematyce są zazwyczaj niewiadomymi, których wartość należy obliczyć. Ponadto ich wartość jest stała, niezmienna, zadana ogólnie poprzez ciąg równań i nierówności. Inaczej jest w informatyce i w programowaniu. Tutaj zmienna jest w zgodzie ze swoją nazwą: jej wartość zmienia się w trakcie wykonywania programu. Zazwyczaj trudno jest przewidzieć, jaka będzie jej wartość na koniec działania programu, a często nas to w ogóle nie interesuje.

Trzecia trudność wynika z braku zrozumienia tego, jak działa komputer i jego pamięć. Nie wiedząc, jak program przechowuje zmienne w pamięci komputera i w jaki sposób się do nich odwołuje, łatwo jest popełnić błąd w rozumowaniu.

Czwarta trudność, która wydaje się być najmniej szkodliwa, wynika z braku dostatecznej znajomości typów danych w danym języku programowania. Trudność ta jednak prowadzi zazwyczaj, w przeciwieństwie do poprzednich, do prostych błędów obliczeniowych, które osoba doświadczona może wyłapać w krótkim czasie. Chodzi tutaj głównie o błędy związane np. z dzieleniem zmiennych typu całkowitego i otrzymaniem także całkowitego wyniku (a nie, jakby to było na matematyce, wyniku rzeczywistego).

Temat zmiennych w programowaniu jest, jak to przedstawialiśmy, tematem trudnym. Daleko mu co prawda do rekurencji, nie mniej jego niedostateczne, a tym bardziej błędne, przyswojenie, może prowadzić do wielu poważnych problemów na dalszym etapie nauki **sztuki** jaką jest programowanie. Dlatego właśnie należy temat ten wyłożyć w sposób możliwie prosty, wręcz łopatologiczny, co staramy się uczynić.

### 3. Zmienne – w najprostszy sposób

Skoro, jak mamy nadzieję udało nam się to pokazać, temat zmiennych nie jest tematem trywialnym i może sprawiać uczniom spore trudności, które mogą wyjść na światło dzienne dopiero po czasie, należałoby go opracować w sposób jak najprostszy i jak najbardziej przystępny dla odbiorcy. Nie jest to zadanie łatwe. Na początek należałoby spróbować zlokalizować źródła trudności tego tematu, co już zrobiliśmy w poprzednim rozdziale. Teraz trzeba się zastanowić, jak te trudności przezwyciężyć. W naszej pracy nie przedstawiamy pełnego i skutecznego rozwiązania, a jedynie pewną propozycję, uwzględniającą w szczególności przezwyciężenie trudności związanych z brakiem zrozumienia istoty działania komputera i programu.

Naszym celem jest przedstawienie uczniom, w sposób jak najprostszy, idei kryjącej się za zarządzaniem i operacjami przeprowadzanymi na pamięci komputera przez program komputerowy. Uważamy, że aby dobrze zrozumieć działanie zmiennych w programowaniu, uczeń najpierw musi zrozumieć, w jaki sposób te zmienne są przez program zapamiętywane. W tym celu opracowaliśmy ćwiczenie – a tak naprawdę zestaw ćwiczeń, opartych na tym samym środowisku, które opiszemy w tym rozdziale.

Wszystkie ćwiczenia tutaj przedstawione mają wspólną cechę – odbywają się bez udziału komputera. Działanie to jest celowe i uzasadnione. Zanim uczeń usiądzie do komputera, by pisać własny program, powinien najpierw dobrze zrozumieć teoretyczne podstawy operacji, którymi zamierza się posługiwać. To, jak bardzo jest to ważne, i że można skutecznie uczyć programowania bez komputera, pokazuje [1].

### 3.1. Ćwiczenie – opis

Ćwiczenie polega na zaprezentowaniu działania pamięci programu w oparciu o prosty, fizyczny model pudełek. Pudełka, a właściwie jedno pudełko z przegródkami, przedstawiać ma fragment pamięci programu – każda przegródka reprezentuje jedno pole rejestru. Tak jak w komputerze jedno pole odpowiada jednemu bitowi, tak tutaj jedna przegródka może pomieścić tylko jedną cyfrę. Przegródki są ponumerowane, a numery odpowiadają adresom pamięci. Oczywiście zamiast liczb szesnastkowych zdecydowaliśmy się na proste liczby dziesiętne, zaczynając od liczby jeden.

Celem ćwiczenia jest zapisanie ciągu prostych operacji na zmiennych i przedstawienie ich na przygotowanym, fizycznym układzie pudełek. Korzystając z takiego układu możemy przedstawić takie operacje jak: inicjalizacja zmiennej, przypisanie wartości, inkrementacja, zmiana wartości, zamiana wartości. Możemy także poruszyć bardziej zaawansowane zagadnienia, jak na przykład przepelnienie pamięci, wyciek pamięci, przepelnienie typu zmiennej.

### 3.2. Ćwiczenie – przygotowanie

Przed przystąpieniem do ćwiczeń, musimy przygotować fizyczny model fragmentu pamięci, na którym będziemy pracować. Do tego celu będzie nam potrzebne długie, podłużne pudełko z przegródkami. Przegródek powinno być ok. 10, tak abyśmy mogli przeprowadzić ciekawsze ćwiczenia. Przegródki muszą zostać ponumerowane kolejnymi liczbami naturalnymi, poczynając od jedynek. Numeracja powinna znajdować się po zewnętrznej stronie pudełka, tak aby była widoczna dla uczniów obserwujących przebieg ćwiczenia. Szkic układu, zawierający 4 przegródki, przedstawiony jest na Rysunku 1.



Rysunek 1 Szkic układu pudełek

Wspomniany układ pudełek przedstawia fragment pamięci komputera. Do przedstawienia zmiennych wykorzystamy natomiast plastikowe kubeczki, z przeczepionymi do nich karteczkami, na których napiszemy wartości zmiennych. Taki układ pozwoli nam na swobodne zmienianie wartości zmiennych, a także przenoszenie ich pomiędzy przegródkami. Kubeczki powinny wystawać ponad pudełko, tak aby były dobrze widoczne.

Wartości zmiennych to nie wszystko, co będzie nam potrzebne do wykonania przygotowanych ćwiczeń. Potrzebujemy także etykiet nazw zmiennych, które także zapiszemy na karteczkach i przypniemy do odpowiedniej przegródki – adresu danej zmiennej w pamięci komputera.

Początkowo, przed rozpoczęciem ćwiczenia, stan pudełka z przegródkami może przyjąć dwie formy:

1. Pudełko jest puste. Zakładamy, że w pamięci nic nie zostało jeszcze zapisane. W trakcie wykonywania ćwiczeń będziemy dokładać zmienne oraz kubeczki z wartościami.
2. Pudełko jest wypełnione. W każdej przegródce znajduje się kubeczek z losową wartością od 0 do 9. Taki układ ma pokazać, że na początku działania programu, w przydzielonym fragmencie pamięci mogą znajdować się śmieci – pozostałości po dawnych obliczeniach. W trakcie ćwiczeń będziemy zmieniać wartości na kubeczkach i przydzielać zmiennym miejsce w pamięci.

Wybór początkowego układu zależy od tego, co chcemy pokazać w danym ćwiczeniu. Początkowe ćwiczenia należy zacząć od pierwszego układu. W celu przedstawienia bardziej realistycznego scenariusza, możemy wykorzystać drugi układ.

### 3.3. Ćwiczenie – przebieg

Przebieg każdego ćwiczenia będzie bardzo zbliżony. Zaczynamy od przedstawienia pseudokodu operacji, którą chcemy zasymulować. Następnie przygotowujemy pudełko w wybranym układzie, a potem, linijka po linijce, symulujemy wykonanie kolejnych operacji na pamięci. W kolejnych podrozdziałach przedstawimy przykładowe ćwiczenia i ich wykonanie na układzie pamięci.

#### 3.3.1. Inicjalizacja zmiennych

Zacznijmy od najprostszego ćwiczenia: utworzenia zmiennej i przypisania jej wartości. Pseudokod prezentujący tę operację wygląda następująco:

1.  $a := 15$

Tworzymy zmienną  $a$  i przypisujemy jej wartość 15. Zanim jednak przedstawimy tę operację na układzie pamięci, musimy określić **typ danych** reprezentowany przez zmienną  $a$ . Jest nam to potrzebne, abyśmy wiedzieli, ile miejsca w pamięci powinno zostać przydzielone tej zmiennej. Nie będziemy tutaj zagłębiać się w szczegóły typów danych, nie zamierzamy też przytaczać typów z takich języków jak C++. Nasz typ danych będzie znacznie prostszy: powiemy, że zmienna  $a$  zajmuje dokładnie 3 miejsca w pamięci. Oznacza to, że zmienna  $a$  może reprezentować liczbę maksymalnie 3-cyfrową. Zapiszemy to w następujący sposób, stosując indeks dolny:

$a_3 := 15$



Zamiast losowych wartości możemy także wstawić znaki zapytania, aby pokazać, że nie wiemy, co znajduje się w danym miejscu w pamięci. Następnie wykonujemy pierwszy krok algorytmu, rezerwując przestrzeń w pamięci na zmienną  $a$ . Stan układu po wykonaniu tej operacji przedstawiony jest w Tabeli 3.

**Tabela 3** Zarezerwowanie pamięci na zmienną

Wartości:	8	4	1	0	8	4	1	5	5
Adresy:	1	2	3	4	5	6	7	8	9
Zmienne:	$a_3$								

Zauważmy, że stan pamięci się nie zmienił. Pojawiła się jedynie informacja o przypisaniu zmiennej  $a$  do adresu 1.

Teraz wykonujemy 2 krok algorytmu, odczytując wartość zmiennej  $a$  z naszego układu pamięci. Wartość ta wynosi 841. Należy zwrócić uwagę, że po każdym uruchomieniu programu wartość ta może ulec zmianie.

Dokończmy symulację algorytmu, wykonując krok 3 na pamięci. Wynik tej operacji prezentuje Tabela 4.

**Tabela 4** Przypisanie wartości do zmiennej

Wartości:	0	1	5	0	8	4	1	5	5
Adresy:	1	2	3	4	5	6	7	8	9
Zmienne:	$a_3$								

Wykonując krok 4 odczytujemy wartość zmiennej  $a$  z pamięci, która teraz wynosi 15.

### 3.3.3. Suma dwóch zmiennych

Zapiszmy algorytm obliczenia sumy dwóch zmiennych:

1.  $a_3 := 15$
2.  $b_3 := 30$
3.  $s_3 := a_3 + b_3$
4. *Wypisz  $s_3$*

Ponownie zaczynamy od przygotowania układu pamięci, zaczynając od pustego układu. Symulujemy krok 1 algorytmu. Jego wynik będzie taki sam jak przy ćwiczeniu inicjalizacji zmiennej i prezentuje go Tabela 1. W kroku 2 tworzymy nową zmienną i zapisujemy ją w pamięci zgodnie z zasadą przydzielania kolejnych wolnych miejsc w pamięci. Wynik tej operacji prezentuje Tabela 5.

Tabela 5 Dwie zmienne

Wartości:	0	1	5	0	3	0			
Adresy:	1	2	3	4	5	6	7	8	9
Zmienne:	$a_3$			$b_3$					

Zauważmy, że zmienna  $b$  została zapisana pod adresem 4, ponieważ był to pierwszy jeszcze nie zajęty adres w pamięci. Wykonajmy teraz krok 3. Podczas tej operacji odczytujemy wartości zmiennych  $a$  i  $b$ , dodajemy je do siebie, a wynik zapisujemy w nowej zmiennej  $s$ . Wynik operacji przedstawia Tabela 6.

Tabela 6

Wartości:	0	1	5	0	3	0	0	4	5
Adresy:	1	2	3	4	5	6	7	8	9
Zmienne:	$a_3$			$b_3$			$s_3$		

W kroku 4 odczytujemy z pamięci wartość zmiennej  $s$ , która wynosi 45.

### 3.3.4. Dalsze ćwiczenia

Oczywiście na stworzonym układzie można zaprezentować także inne, bardziej skomplikowane algorytmy. Przytoczone ćwiczenia mają służyć za intuicję kryjącą się za samym pomysłem. Stworzony układ możemy wykorzystać do zaprezentowania takich operacji jak np.:

- Zamiana zmiennych – algorytm trzeciej ręki
- Instrukcja warunkowa
- Pętla warunkowa
- Pętla iteracyjna
- Operacje na tablicach, listach, stosach, kolejkach i innych strukturach

Jak zostało już wcześniej wspomniane, układ można wykorzystać także do zaprezentowania zagadnień związanych z pamięcią, takich jak limit pamięci. Można także na nim zaprezentować zmienne tymczasowe i zwalnianie pamięci. Raz stworzony układ może towarzyszyć uczniom przez dłuższy okres ich edukacji programistycznej, nie trzeba ograniczać się jedynie do zaprezentowania samych zmiennych.

## 4. Wnioski i dalsze plany

Zaprezentowaliśmy szkieletowy opis ćwiczenia, które ma umożliwić uczniom zdobycie intuicji na temat działania pamięci komputera i programu, dzięki czemu łatwiej im będzie poprawnie zrozumieć i wykorzystywać zmienne w programowaniu. Układ ma



służyć jako swoisty symulator pamięci komputera, a jego niewątpliwą zaletą jest fakt, że do jego użycia nie jest potrzebny komputer. Jak zostało to wspomniane w poprzednim rozdziale, układ można wykorzystać do zaprezentowania wielu istotnych zagadnień programistycznych.

W przyszłości planujemy rozwinąć przedstawiony pomysł, wzbogacając go o kolejne ćwiczenia i przykłady. Być może układ posłuży także do opracowania takich tematów jak wprowadzenie do pętli, czy też wprowadzenie do struktur danych.

Na koniec pragniemy także wspomnieć, że temat tutaj przedstawiony stanowi fragment otwartego projektu pod nazwą *Trudne Tematy w Najprostszy Sposób*, w skład którego wchodzi takie tematy jak *Wprowadzenie do algorytmów*, *Wprowadzenie do funkcji*, czy też *Wprowadzenie do rekurencji*. Wszystkie materiały można znaleźć pod [4].

## Literatura

Pozycje w wykazie literatury podajemy alfabetycznie. Przykładowe opisy ilustrujemy poniżej. W tekście odwołujemy się do pozycji literatury umieszczając jej numer w nawiasach kwadratowych.

1. Computer Science Unplugged <https://csunplugged.org/en/>
2. Podstawa programowa, szkoła podstawowa I-III  
<https://podstawaprogramowa.pl/Szkola-podstawowa-I-III>
3. Podstawa programowa, szkoła podstawowa IV-VIII  
<https://podstawaprogramowa.pl/Szkola-podstawowa-IV-VIII>
4. Trudne Tematy w Najprostszy Sposób, *Damian Kurpiewski, Krzysztof Skowronek*  
<https://github.com/blackbat13/CS-HTiEW>