

Koncepcja podręczników z serii „Teraz bajty” przeznaczonych do realizacji informatyki w zakresie podstawowym w klasach I-III szkół ponadpodstawowych

Autor: Grażyna Koba

W skład serii wchodzi podręczniki:

- *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres podstawowy. Klasa I*
- *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres podstawowy. Klasa II*
- *Teraz bajty. Informatyka dla szkół ponadpodstawowych. Zakres podstawowy. Klasa III*



1. Wprowadzenie

W podręcznikach do realizacji informatyki w zakresie podstawowym materiał nauczania dostosowany jest do poziomu szkoły ponadpodstawowej.

Uczniowie przez trzy lata będą poznawać metody rozwiązywania problemów z wykorzystaniem komputera i oprogramowania komputerowego. Będą to problemy algorytmiczne, do których rozwiązania będą wykorzystywać środowiska programistyczne (moduł C), i inne problemy, które będą rozwiązywać, stosując oprogramowanie użytkowe (moduły B i D), a także różne urządzenia cyfrowe (Moduł A). W trakcie nauki informatyki będą rozwijać **myślenie komputacyjne**.

2. Myślenie komputacyjne – sposób realizacji

W serii „Teraz bajty” punktem wyjścia do realizacji celu, jakim jest „rozwój umiejętności myślenia komputacyjnego”, jest definicja podana przez propagatorkę myślenia komputacyjnego – profesor Jeannette M. Wing:

„Myślenie komputacyjne to procesy myślowe angażowane w formułowanie problemu i przedstawianie jego rozwiązań w taki sposób, aby komputer – człowiek lub maszyna – mógł skutecznie wykonać.”

Myślenie komputacyjne może dotyczyć wielu obszarów zastosowań. Jest ono bardzo ważne w innych dziedzinach niż informatyka, w których można angażować procesy myślowe w formułowanie problemów i przedstawianie ich rozwiązania również bez użycia komputera. Natomiast w podręcznikach z serii „Teraz bajty” przyjąłam założenie, że na lekcjach informatyki uczniowie będą przedstawiać rozwiązanie problemu w taki sposób, aby komputer-maszyna mógł je wykonać. Zwłaszcza, że większość treści szczegółowych z podstawy programowej wskazuje właśnie na wykorzystanie komputera do rozwiązania problemu. Na przykład:

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

Uczeń:

2. do realizacji rozwiązań problemów prawidłowo dobiera środowiska informatyczne, aplikacje [...]

3. przygotowuje opracowania rozwiązań problemów, posługując się wybranymi aplikacjami [...]

W podręcznikach do klas I-III systematycznie wprowadzamy rozwijanie myślenia komputacyjnego.

W podręczniku do **klasy I** wyjaśniamy krótko, że w trakcie nauki informatyki będziemy rozwijać myślenie komputacyjne. Nie określamy jeszcze formalnych etapów, ale pokazujemy stosowanie metod wywodzących się z informatyki do rozwiązywania problemów z różnych dziedzin. Na przykład na stronach 155-159 pokazujemy, jak zaprojektować rozwiązanie (tu: problem sumowania n liczb), aby jak najbardziej efektywnie i optymalnie wykorzystywać możliwości komputera (tu: wykonywanie powtarzających się obliczeń na wielu liczbach, czyli rozwiązywanie problemów iteracyjnych). Określamy również niezbędne pojęcia tj. *iteracja*, *pętla*, *zmienna sterująca*. Pokazujemy na przykładzie stosowanie iteracji do sumowania n liczb i wyjaśniamy, jak ją zaimplementować w wybranym języku programowania.

Wielokrotnie zaczynamy od formułowania problemu (przykłady podano w punkcie 3. koncepcji), a następnie omawiamy, jak wykorzystać metody z informatyki do ich rozwiązania.

W podstawie programowej dla zakresu podstawowego myślenie komputacyjne uwzględniono w punkcie 1. treści szczegółowych pierwszego celu kształcenia:

„I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Uczeń:

1. planuje kolejne kroki rozwiązywania problemu, z uwzględnieniem podstawowych etapów myślenia komputacyjnego (określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania).”

Przykłady rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego z podaniem i uwzględnieniem tych etapów przedstawiamy w podręczniku do **klasy II** (str. 55). Podkreślamy również sposób projektowania rozwiązania: „Powinniśmy tak zaprojektować sposób rozwiązania, aby był jak najbardziej efektywny i optymalnie wykorzystywał możliwości komputera”. Zwracamy również uwagę na ważną cechę rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego: „aby doświadczenia nabyte przy rozwiązywaniu jednego problemu mogły zostać wykorzystane przy rozwiązywaniu innych problemów”. Przykładowe problemy w klasie II (przedstawione na stronach 55-59) dotyczą różnych dziedzin i sytuacji życiowych (badanie ruchu na skrzyżowaniu, sprzedaż gazet czy zmiany klimatyczne). Po sformułowaniu problemu, określeniu modeli i pojęć oraz analizie problemu pokazujemy dobór odpowiedniego narzędzia do jego rozwiązania (tu: arkusz kalkulacyjny). Wskazujemy też drogę do znalezienia rozwiązania, przykład zaprogramowania i testowania rozwiązania oraz wykorzystanie poznanego rozwiązania do rozwiązania innego podobnego problemu, np. jak rozwiązać problem „Sprzedawcy gazet” z wykorzystaniem rozwiązania problemu „Skrzyżowanie ulic” (str. 65). Ponadto polecamy uczniowi (w ćwiczeniu 22., str. 65) samodzielne określenie problemu i przygotowanie jego rozwiązania.

W **klasie III** uczniowie rozwiązują trudniejsze problemy z wykorzystaniem poznanego w klasie II sposobu postępowania z uwzględnieniem etapów myślenia komputacyjnego, bez ich szczegółowego wyodrębniania – uznałam bowiem, że ten sposób myślenia powinien już być dla ucznia naturalny i płynny.

Oto kilka wybranych przykładów.

Zaczynamy od problemu, np. utworzenia bazy danych do obsługi zawodów pływackich (str. 42). Określamy rzeczywisty model (podobnie jak to robiliśmy w klasie II), przedstawiając zasady przeprowadzania takich zawodów w rzeczywistości (na potrzeby podręcznika – model jest uproszczony). Określamy również pojęcia niezbędne do rozwiązania problemu, m.in.: *rekord, pole, relacja*.

Podczas badania rzeczywistej sytuacji zwracamy uwagę na to, jakie dane są gromadzone i w jaki sposób przetwarzane. Pokazujemy następnie, jak dane powinny być zaprezentowane, aby było możliwe przetwarzanie ich za pomocą komputera. Zatem przedstawiamy rozwiązanie w taki sposób, aby komputer mógł je skutecznie wykonać – w tym przypadku otrzymać odpowiednie zestawienia, np. uporządkowane według najlepszego czasu wyniki danych zawodów pływackich. Na koniec uczniowie testują rozwiązanie (tu: bazę danych o zawodach pływackich na przykładowych fikcyjnych danych).

Uczniowie muszą wykazać się również kreatywnością i, korzystając z doświadczeń nabytych podczas rozwiązywania przykładowego problemu, rozwiązują inny, podobny problem (tu: utworzenia bazy danych do obsługi gabinetu stomatologicznego). Samodzielnie „przechodzą” wszystkie etapy myślenia komputacyjnego. Muszą określić rzeczywisty model (tu: wymagania bazy danych) i zaprogramować rozwiązanie tak, aby komputer mógł je skutecznie wykonać. Na koniec również testują rozwiązanie, wprowadzając przykładowe dane.

W przypadku problemów z dziedziny algorytmiki, np. szyfrowania wiadomości, zaczynamy również od określenia problemu (str. 123). Zanim zdefiniujemy pojęcia związane z szyfrowaniem i modele (tu: schematy) szyfrowania, zachęcamy uczniów do samodzielnego odkrywania schematów szyfrowania. Uczniowie stosują ten sam algorytm (tu: szyfrowanie przestawieniowe) do rozwiązania innego problemu (tu: tworzenia anagramów). Następnie uczniowie tylko niezbędne wskazówki, aby mógł zaprogramować dany algorytm szyfrowania w wybranym języku.

Rozwijanie myślenia komputacyjnego występuje również w tematach z grafiki, np. podczas wykonania projektu 3D ogrodu. Tworzymy model ogrodu na kartce (str. 85). Projektujemy rozwiązanie (tu: projekt ogrodu) tak, aby można było je wykonać na komputerze z wykorzystaniem odpowiedniego oprogramowania (tu: aplikacji do tworzenia modeli 3D). W tym przypadku dodatkowo projekt komputerowy będzie następnie zrealizowany w rzeczywistości. Poza tym uczeń nabywając doświadczenia dotyczące pracy nad tym problemem, wykorzystuje je do zaprojektowania własnego pokoju.

Innym przykładem, np. z robotyki jest wykonanie robota-zabawki, z wykorzystaniem doświadczeń nabytych przy projektowaniu sterowania podlewaniem ogrodu (temat D1).

W każdym przypadku uczeń uczestniczy czynnie w procesie szukania rozwiązania, używając odpowiedniego narzędzia (aplikacji czy środowiska programistycznego).

W podręczniku do klasy III (oraz pozostałych z serii „Teraz bajty” – do klasy I i II) uwzględniamy zatem zapis podany w uwagach o realizacji podstawy programowej, że „Najważniejszym celem kształcenia informatycznego uczniów jest rozwój umiejętności myślenia komputacyjnego, skupionego na kreatywnym rozwiązywaniu problemów z różnych dziedzin ze świadomym wykorzystaniem przy tym metod i narzędzi wywodzących się z informatyki, w tym programowania.”

Natomiast więcej i szerzej na temat myślenia komputacyjnego uczeń będzie dyskutować na informatyce w zakresie rozszerzonym. Zgodnie bowiem z punktem 8. celu I podstawy programowej dla zakresu rozszerzonego:

„I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.

Uczeń:

8. dyskutuje na temat roli myślenia komputacyjnego i jego metod, takich jak: abstrakcja, reprezentacja danych, dekompozycja problemu, redukcja, myślenie rekurencyjne, podejście heurystyczne w rozwiązywaniu problemów z różnych dziedzin.”

3. Podejście metodyczne

Prezentowane w podręczniku metody, pojęcia i przykłady uwzględniają różnorodność oprogramowania, jaka występuje w szkołach. Uczeń poznaje zagadnienia w sposób systematyczny i uporządkowany, a rozwiązując niebanalne zadania, rozwija wyobraźnię twórczą i uczy się logicznego myślenia. Omawiane tematy powinny uczniów inspirować i zachęcać do samodzielnego zgłębiania wiedzy.

Jednym z celów nowej podstawy programowej jest „rozumienie, analizowanie i rozwiązywanie problemów”, dlatego w serii *Teraz bajty* wyodrębniamy w niektórych punktach problemy, który następnie analizujemy i dla których szukamy rozwiązania.

Zgodnie z nową podstawą programową, rozwiązywania problemów należy uczyć przy każdej okazji, również podczas posługiwania się aplikacjami takimi jak edytor grafiki, edytor tekstu, arkusz kalkulacyjny, program do tworzenia prezentacji multimedialnych czy baz danych. Dlatego w podręcznikach z serii *Teraz bajty* są również tematy, w których uczeń poznaje metody tworzenia rysunków, tekstów komputerowych, wykonywania obliczeń w arkuszu kalkulacyjnym, gromadzenia i przetwarzania danych w bazach danych czy tworzenia stron internetowych. Wyodrębnione problemy dotyczą nie tylko programowania, ale i stosowania aplikacji w różnych dziedzinach, np. matematyce, języku polskim, a także w życiu codziennym.

Przykłady (klasa I):

1. „Chcemy napisać program sprawdzający, czy wprowadzona z klawiatury liczba jest dodatnia czy niedodatnia (ujemna albo równa zero). Jak zrealizować ten algorytm w arkuszu kalkulacyjnym? W jaki sposób napisać program realizujący sytuację warunkową w wybranym języku programowania?” – str. 142 podręcznika do klasy I.
2. „Napisaliśmy w edytorze tekstu obszerną pracę z języka polskiego. Jak najszybciej sformatować w jednakowy sposób tytuły rozdziałów i podrozdziałów?” – str. 37 podręcznika do klasy I.
3. „Chcesz przygotować tradycyjną sałatkę jarzynową na imprezę urodzinową. W Internecie są przepisy, ale w proporcjach dla dwóch, pięciu czy dziesięciu osób. Nie możesz znaleźć przepisu dla trzynastu osób – uczestników spotkania. Jak napisać program w języku C++, który obliczy, ile potrzebujemy danego składnika sałatki i wyświetli wynik na ekranie?” – str. 122 i 137 podręcznika do klasy I.

Przykłady (klasa II):

1. „Tabela na rysunku 4. zawiera liczby godzin przepracowanych przez pracowników pewnej firmy w ciągu dwóch tygodni. Chcemy dla każdego pracownika obliczyć płacę w każdym tygodniu oraz mieć możliwość porównywania płacy w pierwszym i drugim tygodniu. Jak to zrobić w arkuszu kalkulacyjnym?” – str. 77 podręcznika do klasy II.
2. „Zbierasz pieniądze na zakup nowego komputera. Wpłacasz co miesiąc dowolną kwotę na swoje konto. Chciesz obliczyć sumę pieniędzy wpłaconych w ciągu roku. Jak zapisać rozwiązanie tego zadania w wybranym języku programowania?” – str. 102 podręcznika do klasy II.
3. „Do magazynu sprzętu RTV dowożone są kilkakrotnie w ciągu miesiąca różne artykuły. Jak napisać program zliczający, ile sztuk każdego artykułu dostarczono we wszystkich dostawach łącznie?” – str. 110 podręcznika do klasy II.

Przykłady (klasa III):

1. „Stomatolog gromadzi informacje o pacjentach i ich kolejnych wizytach na kartach, które wypełnia ręcznie. Chcemy umożliwić mu gromadzenie i przetwarzanie danych z wykorzystaniem komputera. W jaki sposób możemy na przykładzie bazy danych *Zawody* utworzyć bazę danych, której celem będzie przetwarzanie danych o pacjentach gabinetu stomatologicznego?” – str. 50 podręcznika do klasy III.
2. „Napisaliśmy pracę domową z informatyki na temat robotów. Okazało się jednak, że w niektórych miejscach w skrócie „np.” wstawiliśmy na końcu przecinek zamiast kropki. Jak szybko wyszukać i poprawić błędne zapisy?” – str. 102 podręcznika do klasy III.

3. „Jesteś kierowcą samochodu dostawczego i musisz przewieźć 3500 książek z drukarni do magazynu. Książki zapakowano w paczki, które umieszczono na paletach. W jednej paczce mieści się 18 książek, natomiast na palecie – 1620 książek. W samochodzie możesz umieszczać tylko pełne palety lub paczki bądź pojedyncze egzemplarze. Jak sprawnie i szybko zapakować książki do samochodu?” – str. 143 podręcznika do klasy III.

W podręcznikach wiele metod objaśniam za pomocą przykładowego rozwiązania problemu. Czasem podane są kolejne kroki postępowania i pokazane są zrzuty lub rysunki. Następnie podane jest ćwiczenie, które uczeń, po analizie danego przykładu, może wykonać samodzielnie. **W klasach starszych (zwłaszcza w klasie III) uczniowie będą próbowali samodzielnie szukać rozwiązania problemów praktycznych, analizować rozwiązanie i próbować samodzielnie formułować algorytm.**

W podręcznikach zostały wyróżnione pojęcia i metody informatyki określające pewne ogólne sposoby postępowania – niezależne od stosowanego oprogramowania czy jego wersji. Ważne fragmenty tekstu zostały w podręcznikach wyróżnione kluczykiem, np. stosowanie stylu niestandardowego w edytorze tekstu. Metody są tak opisane, aby uczeń mógł je zastosować, korzystając z dowolnej wersji edytora tekstu. Treść ćwiczeń również jest zapisana w sposób umożliwiający ich wykonanie w dowolnym edytorze tekstu lub dowolnej jego wersji. W podobny sposób, jak praca w edytorze tekstu, prezentowane są w podręczniku również inne zagadnienia, m.in. tworzenie grafiki, praca z dokumentem tekstowym i tworzenie prezentacji multimedialnych, algorytmika i programowanie.

Szczególne uwagi zostały zwrócone na podobieństwa metod pracy w różnych wersjach programów komputerowych. Ważne jest, aby uczniowie zorientowali się, że w kolejnej wersji tego samego programu niektóre czynności wykonuje się bardzo podobnie, a często tak samo, na przykład zapisywanie czy otwieranie pliku (zazwyczaj w opcji **Plik** lub **File**), a operacje z wykorzystaniem **Schowka** w opcji **Edycja** lub **Edit**).

Aby utrwalić poznaną metodę, uczniowie powinni wykonać kilka podobnych zadań. Od pierwszych zajęć uczniowie powinni wykonywać dużo ćwiczeń praktycznych oraz zdobywać określoną wiedzę (oczywiście w zakresie podstawy programowej).

Utrwalaniu wiedzy i umiejętności służą też powtórki materiału na początku każdego tematu *Warto powtórzyć*, zebranie w kilku punktach ważniejszych treści z danego tematu *Warto zapamiętać* oraz pytania i zadania utrwalające na końcu każdego tematu.

Powinniśmy zdawać sobie sprawę, że umiejętności uczniów w danej grupie mogą być na różnym poziomie. Dla uczniów bardziej zainteresowanych przygotowane są dodatkowe zadania, umożliwiające im pogłębianie wiedzy, ujęte w podręcznikach jako zadania dla zainteresowanych.

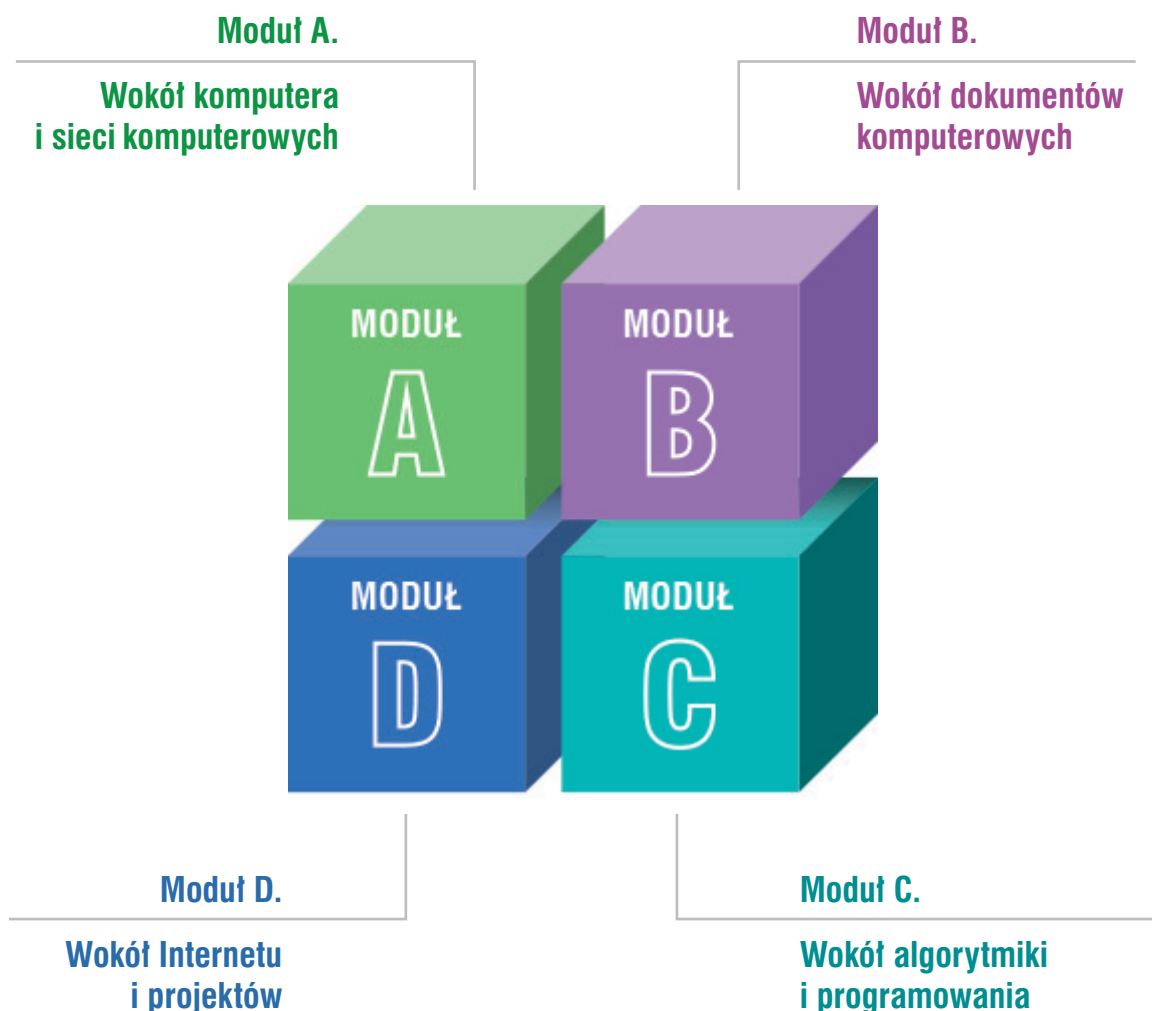
Na każdym poziomie zaplanowałam zadania projektowe, w których uczniowie wykorzystują poznaną wiedzę i zdobyte umiejętności. Ponadto podczas opracowywania projektów, uczniowie zdobywają kompetencje społeczne i uczą się podejmowania indywidualnych i grupowych decyzji, skutecznego działania w grupie oraz umiejętnego planowania pracy.

Prezentowana w podręcznikach metodyka nauczania informatyki umożliwia skuteczne kształcenie informatyczne uczniów, a przede wszystkim zapewnia rozumienie i systematyczne utrwalanie pojęć, metod i procesów związanych z informatyką.

Jeśli uczniowie dobrze poznają i utrwala materiał podstawowy, wówczas będą świadomie wykorzystywać poznane narzędzia i metody informatyki do rozwiązywania problemów z różnych dziedzin. Będą też w stanie wymyślić coś nowego, twórczego. Bardziej kreatywnym jest bowiem człowiek, który ma pewien zasób wiedzy – przynajmniej tej podstawowej.

4. Budowa modułowa podręczników z serii *Teraz bajty*

Każdy podręcznik składa się z czterech modułów o podobnych treściach, a każdy z nich z tematów o wyrazistej strukturze merytorycznej i metodycznej. W podręcznikach jest zachowana spiralskość treści nauczania (czyli „utrwalania wcześniej kształtowanych umiejętności i przemyślanego rozszerzania ich o nowe, odpowiednio do naturalnego rozwoju ucznia”) oraz przyrostowy charakter treści nauczania. W kolejnej klasie w odpowiadających sobie modułach treści są odpowiednio utrwalane i rozszerzane.

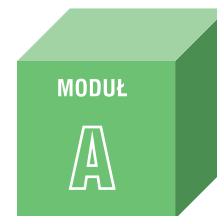


Modułowa budowa podręcznika pozwala realizować tematy w podanym układzie, ale również dostosować kolejność ich realizacji do warunków szkoły i grupy uczniów.

Uwaga: W tytułach modułów mogą wystąpić niewielkie różnice, np. w klasach I i III moduł A ma tytuł „Wokół komputera”, w klasie II – „Wokół komputera i sieci komputerowych”.

W podręcznikach do klasy pierwszej i drugiej uwzględniam fakt, że uczniowie w szkole podstawowej mogli mieć prowadzone zajęcia według różnych programów nauczania, z różnych podręczników, a czasem (co nie jest rzadkością) – bez podręcznika. Dlatego niektóre treści (zwłaszcza te przydatne na poziomie szkoły ponadpodstawowej) najpierw systematyzuję, aby wyrównać poziom przygotowania uczniów, a potem rozszerzam.

Moduł A. Wokół komputera i sieci komputerowych

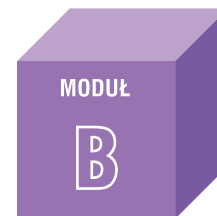


W **klasie I** zaczynamy od przedstawienia pojęć podstawowych. W module A systematyzujemy i rozszerzamy informacje na temat parametrów komputera, czyli co warto wiedzieć, gdy chcielibyśmy kupić komputer. Systematyzujemy i rozszerzamy również zagadnienia związane z urządzeniami peryferyjnymi i oprogramowaniem komputerowym (w tym systemami operacyjnymi). Umieszczenie tych zagadnień w zakresie podstawowym będzie przydatne dla uczniów, którzy będą realizowali podstawę programową z zakresu rozszerzonego, według której uczeń „projektuje rozbudowę i zakup nowego zestawu komputerowego oraz oprogramowania”.

W **klasie II** usystematyzujemy i rozszerzymy wiedzę i umiejętności uczniów w zakresie sieci komputerowych, m.in.: topologie sieci komputerowej, sieci klient-serwer, peer-to-peer, sposoby identyfikowania komputerów w sieci. Szczególną uwagę zwracamy na zasady bezpieczeństwa i ochronę danych w komputerach i sieciach komputerowych (w tym informacji wrażliwych), na uświadamianie sobie konsekwencji łamania zasad.

W **klasie III** wracamy spiralnie do treści wiązanych z komputerem i rozszerzamy wiedzę ucznia na temat budowy komputera i działania procesora. Uczniowie zapoznają się ze sposobem reprezentowania liczb w komputerze (systemem dwójkowym) oraz z zamianą liczb między pozycyjnymi systemami liczbowymi, wykonując praktyczne ćwiczenia. Na koniec zapiszą te algorytmy w wybranym języku programowania (w jednym z tematów modułu C).

Rozszerzamy również informacje o urządzeniach cyfrowych, zwłaszcza o nowoczesnych technologiach (w klasie I omówiliśmy tylko krótko w tabeli urządzenia peryferyjne). Zgodnie z podstawą programową uczniowie zapoznają się z możliwościami różnych urządzeń i towarzyszącego im oprogramowania, dlatego zaproponowałam w temacie opis przykładowych urządzeń o różnorodnym zastosowaniu (nie tylko szkolnym) i różnych rozwiązaniach technologicznych, aby zainspirować uczniów do zainteresowania się nimi i poszukiwania dodatkowych informacji. Na wielu przykładach pokazujemy uczniowi, jak można wykorzystać możliwości urządzeń cyfrowych do rozwiązywania rzeczywistych problemów. Celowo w tym temacie omawiamy drukowanie na drukarce 3D, ponieważ w klasie III uczniowie poznają projektowanie modeli 3D (w module B) i będą już przygotowani do ewentualnego ich drukowania. Celowo omawiany jest przykład ławki ogrodowej, bo taką projektują. Jeśli szkoła nie dysponuje drukarką 3D, nauczyciel będzie mógł wyświetlić uczniom film pokazujący kolejne fazy druku 3D (korzystając z pomocy dydaktycznych przygotowanych dla nauczyciela na stronie wydawnictwa).



W tematach modułu B zawieramy propozycje opracowywania rozwiązań problemów z różnych dziedzin z wykorzystaniem wybranych aplikacji.

W **klasie I** uczniowie opracowują dokumenty tekstowe o rozbudowanej strukturze, posługują konspektem dokumentu, stosują style tekstu, dodają spisy treści, rysunków i tabel. Poznają również pracę nad dokumentem w trybie recenzji oraz praktyczne rady dotyczące edycji tekstu. Uważam, że prawidłowe przygotowywanie dokumentów tekstowych jest ważną umiejętnością, dlatego są w podręczniku dwa tematy poświęcone temu zagadnieniu. W ćwiczeniach uczniowie opracowują gotowe teksty, np. przygotowane przez nauczyciela lub pobrane z Internetu. W zadaniach sprawdzających pracują nad własnym tekstem.

Uczniowie w klasie pierwszej tworzą i edytują projekty w grafice rastrowej, korzystając z programu GIMP. Po zapoznaniu się z różnymi możliwościami tworzą własne projekty, np. fotomontaże inspirowane rysunkami z podręcznika. W ostatnim temacie modułu B uczniowie poznają typy prezentacji, a następnie zapoznają się zasadami przygotowania prezentacji w ten sposób, że analizują tekst i oceniają, czy treść punktu została odpowiednio przedstawiona na slajdzie pokazanym na rysunku w podręczniku. Na koniec zgodnie z tymi zasadami tworzą prezentację na zadany temat. W podsumowaniu wykonują prezentacje na różnorodne tematy, np. o wybranym człowieku ze świata kultury lub sportu, ofertę handlową reklamującą firmę na targach.

W **klasie II** w module B dwa tematy poświęcimy na rozwiązywanie problemów z różnych dziedzin (m.in. matematyki, fizyki, geografii) z wykorzystaniem funkcji i narzędzi arkusza kalkulacyjnego, w tym tabel i wykresów przestawnych. Podajemy przykłady rozwiązywania problemów z wykorzystaniem myślenia komputacyjnego z uwzględnieniem jego etapów tj. określenie problemu, definicja modeli i pojęć, znalezienie rozwiązania, zaprogramowanie i testowanie rozwiązania.

W klasie drugiej uczniowie poznają możliwości nowego programu (Inkscape), w którym wykonują projekty graficzne w grafice wektorowej.

W **klasie III** omawiamy jedno z ważniejszych zagadnień w informatyce – bazy danych. Wychodzimy od problemu gromadzenia informacji i wyjaśniamy jak to, co kiedyś było wykonywane metodami tradycyjnymi, może wykonywać komputer (przetwarzać dane).

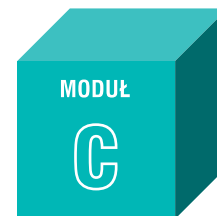
Uczniowie na przykładzie zawodów pływackich, poznają metody wykorzystania komputera do obsługi takich zawodów. Rozwiązują w ten sposób myślenie komputacyjne - dowiadują się, jak prace, które kiedyś wykonywano tradycyjnie (nawet nie sądząc, że może wykonywać je komputer), można wykonywać z wykorzystaniem komputera. Dowiadują się też na czym polega przetwarzanie danych w bazach danych. Następnie doświadczenia zdobyte przy poznawaniu jednego systemu baz danych wykorzystają do wykonania samodzielnie bazy danych dotyczącej innego systemu – tu: obsługi gabinetu stomatologicznego.

Uczniowie po wykonaniu własnej bazy danych powinni zauważyć celowość gromadzenia w ten sposób danych. Jeśli chodzi o posługiwanie się programem do tworzenia baz danych, podajemy tylko niezbędne wskazówki, aby uczniowie mogli utworzyć tabele, formularze, kwerendy, raporty. Wyjaśniamy też niezbędne pojęcia.

Podsumowaniem działu o bazach danych jest połączenie możliwości edytora tekstu i programu do tworzenia baz danych do tworzenia korespondencji seryjnej (etykiet adresowych i listów seryjnych).

W klasie III spiralnie wracamy do projektowania grafiki, rozszerzając zagadnienia o projekty 2D oraz 3D z wykorzystaniem odpowiedniego programu. Zaczynamy od sformułowania problemu – zaprojektowania ogrodu. Pokazujemy w ten sposób, że projekt zrealizowany na komputerze, można potem wykonać w rzeczywistości. Mamy tu więc również elementy myślenia komputacyjnego.

Uczeń zapoznaje się z programem graficznym, projektując model 3D ławki. Podajemy tylko niezbędne wskazówki dotyczące posługiwania się programem. Jeśli szkoła będzie wyposażona w drukarkę 3D, uczniowie będą mogli tę ławkę wydrukować. Celowo pokazujemy tylko kilka elementów projektu ogrodu (nie jest to gotowy projekt), aby uczniowie uzupełnili go lub wykonali według własnego pomysłu. Następnie, korzystając ze zdobytych umiejętności, będą mogli już samodzielnie zaprojektować własny pokój.



Klasę I zaczynamy od usystematyzowania i przypomnienia pojęć podstawowych związanych z algorytmiką i programowaniem. Na początku skupiamy się na formułowaniu sytuacji problemowej i określaniu tylko specyfikacji zadania, ponieważ przedstawianie algorytmów w postaci list kroków i schematów blokowych występują w podstawie programowej dla zakresu rozszerzonego. Na lekcjach informatyki realizowanej w zakresie rozszerzonym uczniowie będą tworzyć programy na podstawie specyfikacji zadania.

Od klasy I jest zachowana zasada spiralnego nauczania i przyrostowego układu treści nauczania. Zgodnie z warunkami realizacji podstawy programowej dla klas VII-VIII opisanymi w rozporządzeniu Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej [...]:

- „Od klasy VII uczniowie, którzy zrealizowali przedmiot informatyka w klasach IV-VI zgodnie z podstawą programową kształcenia ogólnego dla 6-letniej szkoły podstawowej, są wprowadzani do myślenia algorytmicznego, poznają podstawowe pojęcia informatyczne i rozwiązują algorytmicznie wybrane problemy. Stawiają pierwsze kroki w wizualnym **lub** tekstowym języku programowania. Dotychczas zdobyte wiedza i umiejętności informatyczne są rozwijane i poszerzane.”
- Natomiast dopiero „Uczniowie, którzy w klasach IV-VI zrealizowali przedmiot informatyka zgodnie z podstawą programową kształcenia ogólnego dla 8-letniej szkoły podstawowej, zostali wcześniej wprowadzeni do myślenia algorytmicznego, poznając podstawowe pojęcia informatyczne i rozwiązując algorytmicznie wybrane problemy, programując przy tym ich rozwiązania. W związku z powyższym dotychczas zdobyte wiedza i umiejętności informatyczne są rozwijane i poszerzane oraz stawiane są pierwsze kroki w tekstowym języku programowania.”

Uczniowie, którzy od września 2019 roku przychodzą do klasy pierwszej wybranej szkoły ponadpodstawowej należą do tej pierwszej grupy i nie muszą posiadać umiejętności programowania w żadnym tekstowym języku programowania, ponieważ nauczyciel mógł realizować treści nauczania w klasach VII-VIII jedynie w oparciu o języki wizualne.

Jeśli jednak nauczyciel realizował programowanie w oparciu o wybrany język tekstowy (wysokiego poziomu), to najczęściej był to jeden z języków (C++, Python) – rzadko wystąpi przypadek, że uczeń w klasach VII-VIII opanował podstawy programowania w obydwu językach. Poza tym podstawa nie podaje nazwy konkretnego języka programowania. W szczególnym przypadku nauczyciel mógł wprowadzić np. podstawy języka Java.

W każdym więc przypadku uczniowie korzystający z tego podręcznika mają możliwość poznania nowych treści i zdobycia nowych umiejętności – przyrostowo.

Dlatego w pierwszej klasie zaczynamy od początku – od podstawowych zasad programowania w języku wysokiego poziomu. Nawiązujemy oczywiście do wiedzy i umiejętności uczniów z programowania w językach wizualnych, porównując np. rozwiązanie tego samego zadania w czterech środowiskach (Scratch, Baltie, C++ i Python), aby uczniowie zauważyli podobieństwa i różnice w ułożeniu odpowiadających sobie poleceń. Przy okazji omawiania działania instrukcji warunkowej również zwracamy uwagę, że działa ona tak samo we wszystkich środowiskach. Celowo w klasie pierwszej uczniowie poznają tylko jeden rodzaj instrukcji warunkowej, aby lepiej utrwalili jej zastosowania.

Uczniowie przez trzy lata uczą się informatyki, więc nauczanie programowania może przebiegać bardzo systematycznie. Istotne jest utrwalanie podstaw programowania, poprzez wykonywanie wielu podobnych zadań utrwalających poznane zasady. Zadania takie są na końcu każdego tematu. Zanim uczniowie zaczną świadomie stosować poznane instrukcje, dobierać odpowiednio typy danych i programować trudniejsze algorytmy, muszą dobrze utrwalić podstawowe zasady programowania.

W **klasie II** omówimy programowanie algorytmów iteracyjnych, w których nie jest z góry określona liczba powtórzeń. Uczniowie poznają funkcje i tablice (listy). Zastosują poznane zasady programowania (m.in. funkcje i złożone struktury danych) do zaprogramowania wybranych algorytmów: badania pierwszości liczby, obliczanie wartości elementów ciągu Fibonacciego, a także algorytmy porządkowania przez wstawianie i metodą bąbelkową.

W tematach modułu C rozwiązujemy problemy z różnych dziedzin i życia codziennego, m.in. sumowanie i porządkowanie elementów (liczb, dostaw sprzętu RTV – tematy C2 i C3), obliczanie salda przy zakupie komputera (temat C1). Staramy się w ten sposób pokazywać uczniom, że niektóre metody informatyki (tu: tablice, funkcje, algorytmy porządkowania) można stosować do rozwiązania różnych problemów – rozwijamy intuicyjnie myślenie komputacyjne.

W **klasie III** (po poznaniu podstaw algorytmiki i programowania) możemy przejść do omówienia trudniejszych algorytmów i ich zaprogramowania.

Staramy się, zgodnie z etapami myślenia komputacyjnego zacząć również od sformułowania rzeczywistego problemu, np. porównywania znaków, szukania wzorca w tekście, szyfrowania wiadomości, szukania książki na półce, układania stosu książek czy załadunku książek do samochodu, wydawania reszty, następnie szukania sposobu rozwiązania problemu, analizy przykładów tak, aby intuicyjnie dojść do sformułowania algorytmu. Następnie płynnie przechodzimy do zapisania danego algorytmu w wybranym języku programowania, naprowadzając ucznia na zastosowanie odpowiednich instrukcji (np. jeśli był to algorytm iteracyjny). Przy okazji wyjaśniamy uczniom niezbędne pojęcia potrzebne do zaprogramowania algorytmu.

Moduł C zamykamy tematem, w którym wracamy do algorytmów poznanych przez ucznia w module A – zamiany liczb między pozycyjnymi systemami liczbowymi.

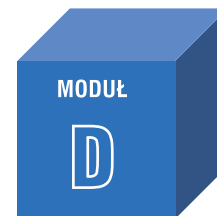
W podręczniku umieszczamy przykładowe funkcje (jako wzorcowe) lub tylko ich fragmenty, aby uczniowie mogli je analizować i stosować w programach. Czasem celowo fragmenty programów są „przykryte”, aby uczniowie mogli je samodzielnie „odkryć”. Przy realizacji każdego algorytmu zwracamy uwagę na testowanie, zwłaszcza jego realizacji w języku programowania.

**W każdej klasie tematy z programowania są tak przygotowane,
aby nauczyciel mógł wybrać realizację podstawy programowej z zakresu programowania
w języku C++ lub Python lub w obydwu tych językach.**

Niezależnie od wyboru, zawsze zostanie zrealizowana podstawa programowa. Zaprezentowanie obydwu języków ma na celu nie tylko pokazanie podobieństw i różnic w składni różnych języków programowania, ale także w realizowaniu tych samych algorytmów. Uczeń, znając już dany algorytm, może zwracać uwagę tylko na jego implementację w danym języku.

Zgodnie z podstawą programową, w wielu ćwiczeniach z programowania umieszczone zostały polecenia, które pomagają uczniom objaśniać przebieg działania programów, testować program na komputerze zgodnie z przyjętymi założeniami i ewentualnie go poprawiać.

Jeśli uczniowie poznają podstawy programowania w języku wysokiego poziomu, będą mogli po ukończeniu szkoły świadomie wybrać dalszy kierunek kształcenia (np. studia informatyczne lub prace w branży informatycznej).



W **klasie I** systematyzujemy pojęcia związane z Internetem, omawiamy i rozszerzamy informacje dotyczące organizacji informacji w Internecie. Uczniowie ćwiczą również zaawansowane sposoby wyszukiwania informacji (co niestety ciągle jest dla nich problemem), np. jak szukać pliku z określonym rozszerzeniem. Wybrane usługi internetowe i inne wydarzenia związane z Internetem pokazujemy na odwróconej piramidzie. Taki obraz może stanowić inspirację do przeprowadzenia np. dyskusji z uczniami na temat różnych usług, tych które są umieszczone na piramidzie i tych których tam nie ma. Uczniowie mogą utworzyć własną piramidę. Inspiracją mogą też być zadania zamieszczone na końcu tematu.

Drugim tematem są wybrane e-usługi. Ich szczegółowe omówienie, zwłaszcza zasad bezpieczeństwa korzystania z nich ułatwi uczniom w przyszłości korzystanie z danej usługi. Treści z tego tematu będą też punktem wyjścia do debaty ZA i PRZECIW, którą uczniowie przeprowadzą w ramach zadania projektowego w ostatnim temacie.

W **klasie II** systematyzujemy różne formy komunikacji, zwracając szczególną uwagę na nowe formy. Cały temat poświęcimy też przepisom prawa (zwłaszcza korzystanie z cudzych materiałów, ochrona wizerunku itp.). Zwrócimy też szczególną uwagę na bezpieczeństwo w sieci i różne zagrożenia. Ponadto uczniowie będą tworzyć stronę internetową wzbogaconą tabelami, listami, elementami dynamicznymi, będą posługiwać się arkuszem stylów, korzystać z oprogramowania i serwisów przeznaczonych do tworzenia stron oraz publikować własną stronę w Internecie.

Ostatnim tematem w każdej klasie są zadania projektowe, w których uczniowie będą rozwijać kompetencje społeczne i wykazywać się aktywnością i kreatywnością. Tematy projektów stanowią podsumowanie zagadnień omawianych w danej klasie. Uczniowie, znając materiał powinni wykazać się kreatywnością w ciekawym, innym, nowym ujęciu danego zagadnienia

W **klasie III** w ramach zadań projektowych omawiamy elementy robotyki, nawiązując do zagadnień omawianych w podręczniku, np. projektowanie ogrodu. Tu pokazujemy, jak można sterować... automatycznie jego podlewaniem. Mamy znów nawiązanie do myślenia komputacyjnego – zamiast podlewać konewką, pokazujemy, jak można wykorzystać do tego celu mikrokontroler i komputer (my możemy w tym czasie usiąść na ławce wydrukowanej na drukarce 3D w zaprojektowanym samodzielnie ogrodzie). W razie problemów z praktycznym wykonaniem tego modelu nauczyciel będzie mógł pokazać film (korzystając z przygotowanych dla niego pomocy dydaktycznych).





Korzystając z doświadczeń podczas tworzenia pierwszego projektu, uczniowie realizują jeszcze kilka, m.in. system „zamknij okno” w pokoju (może w tym, który sami zaprojektowali), a na koniec własnego robota-zabawkę.

W ostatnim temacie spiralnie wrócimy do tworzenia stron internetowych, skupiając się na wzbogacaniu stron różnymi nowymi elementami. Zwracamy większą uwagę na projektowanie elementów graficznych na stronie – na estetykę strony. Korzystając z podanych przykładów, uczeń na zakończenie wykorzystuje te elementy, przygotowując projekt własnej strony.

5. Rozkład treści nauczania w podręcznikach do klas I-III z serii *Teraz bajty*





Teraz bajty.
Informatyka dla szkół ponadpodstawowych.
Zakres podstawowy.
Klasa I



	Moduł A Wokół komputera [3 godz.]	
	Temat A1. Komputer i urządzenia peryferyjne	2
	Temat A2. Systemy operacyjne i inne oprogramowanie	1
	Moduł B Wokół dokumentów komputerowych [9 godz.]	
	Temat B1. Opracowywanie dokumentów tekstowych o rozbudowanej strukturze	2
	Temat B2. Dzielenie dokumentu tekstowego i praca w trybie recenzji	2
	Temat B3. Opracowywanie grafiki rastrowej	3
	Temat B4. Zasady tworzenia prezentacji multimedialnej	2
	Moduł C Wokół algorytmiki i programowania [14 godz.]	
	Temat C1. Wprowadzenie do programowania i rozwiązywania problemów z wykorzystaniem komputera	2
	Temat C2. Tworzenie programów w języku C++	3
	Temat C3. Tworzenie programów w języku Python	3
	Temat C4. Stosowanie instrukcji warunkowych w językach C++ i Python	3
	Temat C5. Stosowanie instrukcji iteracyjnych w językach C++ i Python	3
	Moduł D Wokół Internetu i projektów [4 godz.]	
	Temat D1. Internet jako „ocean informacji”	2
	Temat D2. Korzystanie z wybranych e-usług	1
	Temat D3. Zadania projektowe	1





**Teraz bajty.
Informatyka dla szkół ponadpodstawowych.
Zakres podstawowy.
Klasa II**



	Moduł A Wokół komputera i sieci komputerowych [4 godz.]	
	Temat A1. Sieci komputerowe	2
	Temat A2. Bezpieczeństwo i ochrona danych w komputerach i sieciach komputerowych	2
	Moduł B Wokół dokumentów komputerowych [8 godz.]	
	Temat B1. Formuły, funkcje i wykresy w arkuszu kalkulacyjnym	3
	Temat B2. Filtry oraz tabele i tabele przestawne w arkuszu kalkulacyjnym	2
	Temat B3. Opracowywanie grafiki wektorowej	3
	Moduł C Wokół algorytmiki i programowania [12 godz.]	
	Temat C1. Więcej o stosowaniu instrukcji iteracyjnych w językach C++ i Python	3
	Temat C2. Stosowanie funkcji w językach C++ i Python	3
	Temat C3. Stosowanie tablic w języku C++ i list w języku Python	3
	Temat C4. Wybrane algorytmy na liczbach naturalnych i algorytmy porządkowania	3
	Moduł D Wokół Internetu i projektów [6 godz.]	
	Temat D1. Wybrane przepisy prawa dotyczące technologii informacyjno-komunikacyjnych	2
	Temat D2. Komunikacja i wymiana informacji w Internecie	2
	Temat D3. Zadania projektowe	2

**Teraz bajty.
Informatyka dla szkół ponadpodstawowych.
Zakres podstawowy.
Klasa III**



	Moduł A Wokół komputera [4 godz.]	
	Temat A1. Reprezentacja danych w komputerze	2
	Temat A2. Wybrane urządzenia cyfrowe	2
	Moduł B Wokół dokumentów komputerowych [10 godz.]	
	Temat B1. Tworzenie baz danych	3
	Temat B2. Tworzenie i stosowanie formularzy w relacyjnej bazie danych	2
	Temat B3. Wykonywanie podstawowych operacji na relacyjnej bazie danych	2
	Temat B4. Projektowanie modeli dwuwymiarowych i trójwymiarowych	3
	Moduł C Wokół algorytmiki i programowania [11 godz.]	
	Temat C1. Algorytmy na tekstach w językach C++ i Python	3
	Temat C2. Algorytmy szyfrowania w językach C++ i Python	2
	Temat C3. Metoda połowienia, podejście zachłanne i rekurencja	3
	Temat C4. Algorytmy na liczbach w językach C++ i Python	3
	Moduł D Wokół Internetu i projektów [5 godz.]	
	Temat D1. Elementy robotyki – projekty	3
	Temat D2. Więcej na temat tworzenia stron internetowych	2