

Wstęp

Tradycyjny program nauczania, realizowany w ramach formalnej edukacji szkolnej w systemie klasowo-lekcyjnym, stanowi propozycję treści i metod realizacji, przy założonych celach i oczekiwanych efektach kształcenia. Taki program musi wypełniać zapisy Podstawy programowej konkretnego przedmiotu i etapu edukacyjnego, pozostawiając niezbyt wiele swobody nauczycielom co do własnej inicjatywy w zakresie szczegółowych tematów. Jeśli nauczyciel będzie uważał za istotne, aby uwzględnić oczekiwania i zainteresowania uczniów, może stanąć przed dylematem: – Czy tworzyć i realizować program oparty na Podstawie programowej, czy jednak bardziej dostosować go do preferencji uczniów i własnych? W rzeczywistości polskiej szkoły wielu nauczycieli wybiera wariant pierwszy. Taki stan rzeczy wpływa negatywnie zarówno na zaangażowanie w proces dydaktyczny i satysfakcję samego nauczyciela, jak też i uczniów, którzy coraz częściej wybierają alternatywne, pozaszkolne źródła wiedzy.

W czasach olbrzymiej roli mediów i nowych technologii informacyjno-komunikacyjnych (TIK) w kształceniu nie da się oddzielić efektów edukacji szkolnej od pozaszkolnej. Dzięki sieci i technologiom mobilnym proces uczenia się jest wszechobecny (*ubiquitous learning*). Rodzi się silna potrzeba tworzenia programów i podstaw metodycznych do efektywnego korzystania z mieszanych form edukacyjnych (*blended learning*), których istotą jest połączenie procesu nauczania stacjonarnego i uczenia się zdalnego. Innowacyjne metody i zróżnicowane ścieżki kształcenia z wykorzystaniem nowych technologii objęte są priorytetem w programach unijnych. Świadczy to o olbrzymiej roli wypracowania skutecznych strategii wdrażania komplementarnych form kształcenia bezpośredniego i pośredniego poprzez Internet. Takim właśnie projektem jest „Strategia Wolnych i Otwartych Implementacji” (SWOI), której istotnym elementem jest prezentowany tu Program nauczania-uczenia się infotechniki, wykorzystujący dedykowane narzędzia: Serwis edukacyjno-społecznościowy e-Swoi i Szkolny Remiks Ubuntu (system i oprogramowanie).

Kompleksowa Strategia edukacyjna i Program nauczania-uczenia się – poprzez mieszane formy edukacji, atrakcyjne treści programowe, innowacyjne metody oddziaływań, wzorcowe narzędzia pracy trenerów, wolne i otwarte środowiska systemowe oraz narzędzia pracy ucznia – animuje i moderuje następujące **założenia taktyczne i realizacyjne**:

- » samodzielne lub zespołowe tworzenie implementacji, tj. projektowanie, programowanie lub konstruowanie jako forma wyrównywania szans i aktywizacji poprzez wykonywanie cząstkowo-

- wych zadań na miarę osobistych zdolności, jako alternatywa dla szkolnego nauczania głównie w zakresie posługiwania się technologiami informacyjno-komunikacyjnymi;
- » formowanie świadomości i pozytywnych postaw uczniów wobec wartości takich, jak: twórczość, aktywizacja i partycypacja, szacunek i partnerstwo, wolność i otwartość, dialog i negocjacja – ważnych w wychowaniu i socjalizacji oraz niezbędnych do budowy filarów pod indywidualne lub zespołowe tworzenie implementacji;
 - » harmonizowanie oddziaływań na sfery poznawcze, emocjonalne i psychomotoryczne, poprzez dobór właściwych proporcji pomiędzy zakresem i trudnością materiału stanowiącego treść kształcenia a formami ćwiczeniowymi i emocjonalno-motywacyjnymi, w tym także zabawowymi (np. tematyka gier logicznych czy graffiti).
 - » wielotorowy, międzypokoleniowy i międzyrówieśniczy transfer wiedzy i umiejętności, zarówno w formach stacjonarnych, jak też w formie *on line*, jako strategia wzmocnienia roli społecznych interakcji edukacyjnych i uzupełnienia tradycyjnej ścieżki nauczyciel → uczeń;
 - » czynnościowe kształtowanie postrzegania i rozumienia kluczowych pojęć informatycznych i mechatronicznych, poprzez wykonywanie atrakcyjnych zadań implementacyjnych, egzemplifikujących dane pojęcia, zamiast wyuczania reguł encyklopedyczno-definicyjnych;
 - » rozwój umiejętności twórczego posługiwania się oprogramowaniem wolnym i otwartym, jako efektywny sposób kształtowania kompetencji infotechnicznych, zamiast dotychczasowego szkolenia w zakresie obsługi drogich, nadmiarowych, zawierających błędy i wręcz niebezpiecznych zamkniętych systemów operacyjnych oraz programów użytkowych;
 - » upowszechnienie umiejętności prowadzenia elektronicznej dokumentacji e-Portfolio, potwierdzającej długofalowy proces samorozwoju i osobisty dorobek uczniów – wynikający z ich partycypacji w projektowaniu i realizowaniu implementacji programistycznych bądź mechatronicznych.

W prezentowanym tu Programie wyraźnie podkreśliłyśmy, iż nauczanie i uczenie się należy zawsze w kształceniu traktować jako elementy nierozłączne. Chodzi o zaakcentowanie w pełni kontekstu semantycznego, gdyż słowo ‘nauczanie’ w języku polskim wskazuje raczej na transfer od nauczyciela do ucznia. Także samo słowo ‘kształcenie’ nie odzwierciedla jednoznacznie tego, co zawarte jest w jego naukowej definicji, a mianowicie – jako integralnego procesu nauczania-uczenia się. Pojęcie ‘nauczanie-uczenie się’ jest ugruntowane w języku środowisk naukowych i oświatowych. Rzadziej jednak występuje jako doprecyzowanie tego, czym jest konkretny program związany z kształceniem. Wynika to stąd, iż łatwiej jest opracować szczegółowe zalecenia *jak nauczać*, niż *jak się uczyć*.

W systemie oświaty dominuje wciąż słowo ‘nauczanie’, mimo że w dokumentach regulujących funkcjonowanie placówek coraz bardziej cele szczegółowe wyrażane są jako stany osiągnięć uczniów. Przykładowo, w Podstawach programowych nadal używane są śródtytuły „Treści nauczania”, mimo że wymagania szczegółowe zoperacjonalizowano jako to, co uczeń powinien umieć wykonać. Aby jednak potrafił to coś wykonać, nie wystarczy próbować go tego nauczyć – to on sam musi podjąć systematyczne działania, aby się tego uczyć. Właśnie z powyższych względów prezentowany tu model jest określony mianem **Programu nauczania-uczenia się**. Pragniemy w ten sposób podkreślić wagę potrzeby rzeczywistego zaangażowania w proces dydaktyczny obu podmiotów, zarówno nauczycieli, jak i uczniów.

Upowszechnienie niniejszego Programu ma na celu wdrożenie taktyki prowadzącej do zmian jakościowych w nauczaniu i uczeniu się podczas pozalekcyjnych form zajęć informatycznych oraz mechatronicznych (nazywanych tu *infotechnicznymi* i oznaczanych skrótem IT). Wyjaśnijmy, że w publikacji przyjęliśmy zasadę skrótowego określania łącznego obszaru merytorycznego z dziedzin komputerowego przetwarzania informacji, inżynierii programowania i technik budowy interfejsów mikroprocesorowych.

Wprowadzone pojęcie ‘infotechnika’ obejmuje tu w szczególności problematykę projektowania cyfrowych implementacji, tworzenia oprogramowania i konstruowania układów elektronicznych. W odróżnieniu od tematyki realizowanej w systemie kształcenia ogólnego, skierowanej głównie na użytkowanie technologii informacyjno-komunikacyjnych (TIK), propagowane tu zajęcia pozalekcyjne z infotechniki (IT) mają na celu przede wszystkim formowanie **kompetencji twórczych** w tych obszarach.

Założenie o potrzebie zmian jakościowych w edukacji infotechnicznej wynika z autentycznego zapotrzebowania pracodawców. Okazuje się, że tylko nieliczni absolwenci szkolnictwa zawodowego, a nawet uczelni, potrafią wykonać twórcze zadania implementacyjne, jakie są wymagane w zakładzie pracy na stanowisku informatyka czy mechatronika. Potrzebne jest inne podejście do edukacji w tym zakresie. Pozostawiając szkolnictwu ogólnemu zadania powszechnego przygotowania wszystkich uczniów do posługiwania się technologiami i aplikacjami użytkowymi, wdrożyć trzeba system odpowiednio wczesnego, uzupełniającego formowania poszerzonych kompetencji infotechnicznych na zajęciach pozalekcyjnych. Proces ten nie może obejmować wyłącznie uczniów wybitnie uzdolnionych, lecz powinien zawierać mechanizmy zainteresowania i wspierania każdego chętnego ucznia i uczennicy.

Model kół zainteresowań infotechnicznych powinien różnić się od dotychczasowego następującymi **elementami innowacji**:



Dominuje na tradycyjnych kółach zainteresowań IT

- Organizowanie kół jedynie dla uczniów, którzy wykazują predyspozycje, bez szerszej promocji korzyści płynących z tej formy zajęć
- Aktywizacja i doskonalenie jednostek uzdolnionych kierunkowo
- Nastawienie na osiągnięcie głównie celów poznawczych i umiejętności (wiedza episteme i techné)
- Taksonomia celów ogólnych ABC, wyznaczająca poziomy zapamiętania, zrozumienia i zastosowania
- Zaprogramowany na realizację celów styl prowadzenia zajęć
- Ocenianie sprawdzające osiągnięcie celów założonych w planie kół

Zalecane w innowacyjnej realizacji kół zainteresowań IT

- Aktywna forma zainteresowania dziedziną IT i nabór na koła szerszej rzeszy uczniów poprzez prezentację promocyjną
- Aktywizacja i wczesne ukierunkowanie grup wymagających wsparcia
- Formowania bardziej trwałych cech kierunkowych, wolijonalnych, świadomości i postaw (wiedza phronesis, metiers i know how)
- Taksonomia dedykowana na potrzeby kół IT, z harmonizowaniem efektów poznawczych, doznaniowych i czynnościowych
- Adaptacyjny styl prowadzenia zajęć, nastawiony na realne możliwości
- Ocenianie rzeczywiste działań i efektów bieżących (wytworów) oraz odroczonego (e-portfolio, dystansowe pomiary zmian cech)

Tworzone ad hoc narzędzia oceniania jakości zajęć i ewaluacji efektów	Standaryzowane narzędzia oceniania jakości zajęć i ewaluacji efektów
Poszukiwanie pomysłów na treści i zadania, tworzenie bądź stosowanie dostępnych konspektów o niespójnych strukturach	Stosowanie przetestowanego, kompleksowego pakietu konspektów w układzie alternatywnych modułów i bloków do wyboru
Stosowanie systemów operacyjnych, narzędzi, instrukcji i materiałów dydaktycznych obwarowanych zamkniętymi licencjami, co utrudnia możliwość kontynuacji samokształcenia w domu	Przekazanie mobilnego, wolnego i otwartego systemu z dedykowanym pakietem narzędzi, źródeł, instrukcji i materiałów dydaktycznych, z możliwością ich doskonalenia i nieograniczonego wykorzystywania
Klasyczna metoda projektu, w której dopuszczalny jest brak założonych rezultatów i ew. kontynuacja na kolejnych zajęciach	Metoda kompletnych dzieł, która zakłada działającą implementację jako konieczny rezultat każdej jednostki zajęciowej
Zadania polegają na rozwiązywaniu problemów algorytmicznych, informatycznych, matematycznych bądź mechatronicznych	Zadania polegają na tworzeniu w pełni funkcjonalnych implementacji (gry, animacje, prezentacje, interfejsy, układy pomiarowe)
Implementacje nastawione są na realizację algorytmów oraz stosowanie funkcji, procedur i instrukcji danego języka	Implementacje wspomagają zrozumienie poprzez wizualizację efektów realizacji danego algorytmu, funkcji, procedury lub instrukcji
Większość operacji na obiektach abstrakcyjnych	Przejsie od reprezentacji enaktywnych i ikonicznych do abstrakcji
Tendencje do nadmiernego czasowo, wyłącznie werbalnego wprowadzania w tematykę zajęć i w treść zadań do wykonania	Skondensowana zajawka wizualna, działaniowa, ew. słowna, pełniąc rolę zwiastuna tematyki i przybliżająca istotę zadań do wykonania
Pełne wyjaśnianie wszystkiego, co z punktu widzenia nauczyciela należałoby przedstawić uczniom	Redukcja objaśnień, sprowadzona głównie do odpowiedzi na pytania inicjowane przez uczniów (responsywność)
Nadmiarowe treści zajęć, co skutkuje nieefektywnym zagospodarowaniem czasu zajęć pozalekcyjnych	Szybkie przemykanie przez treść, z celowym niedostatkim informacji możliwych do pozyskania samodzielnie (zapping).
Podpowiadanie /podawanie gotowych rozwiązań problemów, jakie powinny być z niewielkim wsparciem pokonane przez ucznia	Wspieranie niewyręczające, np. pytania naprowadzające na podjęcie właściwej decyzji i rozwiązanie problemu przez ucznia (inquiring)
Przewaga pojęciowego sterowania czynnościami uczniów	Przewaga czynnościowego kształtowania pojęć.
Programowanie imperatywne – tworzenie implementacji głównie poprzez czysto tekstowe pisanie kodu źródłowego	Programowanie wizualne – tworzenie implementacji w środowisku GUI poprzez wykorzystanie gotowych kontrolerek (widżetów)
Uczenie się programowania jako od razu próba pisania kodu źródłowego, zamiast najpierw czytania kodu ze zrozumieniem	Uczenie się programowania w cyklu: czytanie kodu, uzupełnianie luk, modyfikacja, próba tworzenia kodu (metoda glottodydaktyczna)
Ciągle w toku zajęć oddziaływania nauczycieli, a zbyt rzadkie pozostawienie uczniom swobody w ich procesach myślowych	Stopniowe redukowanie w toku zajęć interakcji z trenerem i z grupą na rzecz pełnego zanurzenia się w procesie twórczym (immersja)
Wdrażanie do rozwiązywania zadań na potrzeby własnego rozwoju	Wdrażanie do upowszechniania i dokumentowania swych osiągnięć w e-repozytorium i e-portfolio
Lokalna współpraca i transfer wiedzy	Ogólnopolska współpraca i transfer
Formy stacjonarne zajęć, wzbogacane rozproszonymi zasobami Internetu	Integracja formy stacjonarnej z samokształceniem zdalnym poprzez zasoby i funkcjonalności dedykowanej platformy e-Swoi

Uczenie się twórczości infotechnicznej jest procesem znacznie bardziej złożonym od nauki użytkowania technologii i niedającym się w pełni zunifikować. Ujmowanie tak zawiłego procesu w ramy konkretnego Programu sprawia duże trudności. Potencjalnym rozwiązaniem tego problemu jest wprowadzenie mechanizmów przygotowujących młodych adeptów do samokształcenia i ukierunkowujących ten proces. Kształtowanie umiejętności rozwiązywania problemów inżynierskich musi koniecznie odbywać się poprzez działania praktyczne, z niewyręczającym wsparciem trenera i rówieśników.

W początkowym etapie wdrażania do nowych obszarów wiedzy infotechnicznej potrzebne jest uwzględnianie aspektów psychopedagogicznych. Wynika to z niepełnej gotowości uczniów rozpoczynających naukę w gimnazjum do procesów umysłowych wyższego rzędu, stąd sposób postępowania można przyrównać do zintegrowanej strategii uczenia wczesnoszkolnego. I tak – przygotowanie do działań na obiektach abstrakcyjnych, jakie występują przy programowaniu, musi być wspierane poprzez operowanie na obiektach rzeczywistych lub symulowanych wizualnie. Uczenie „wypowiadania się” w kodzie języka programowania musi być poprzedzone wdrożeniem najpierw do odczytywania elementów składni przykładowego kodu (instrukcji), a dopiero później pisania większych konstruktów (procedur, funkcji).

Na kołach zainteresowań harmonizowane muszą być aspekty poznawczo-kształcące, emocjonalno-motywacyjne i psychomotoryczne, z waloryzacją trudnych treści i formowaniem cech względnie trwałych, przynoszących korzyści w dłuższej perspektywie. Służy temu wypracowana i przedstawiona tu struktura realizacji jednostek dydaktycznych w formie kół, z **metodyką szczegółową i taksonomią efektów**. Zaleca się, aby na każdym zajęciach osiągnąć cztery fazy: *sensytywności* (uwrażliwienie), *responsywności* (uaktywnienie), *problemowości* (decydowanie) i *konstruktywności* (tworzenie). Odbywa się to poprzez uspójnianie wielu metod nauczania-uczenia się i rodzajów oddziaływań, formowanie różnych komponentów postaw i osiągnięcie kierunkowych efektów świadomościowo-emocjonalnych.

Dedykowany na koła zainteresowań Program nauczania-uczenia się infotechniki doprecyzowany jest przede wszystkim w **Konspektach-scenariuszach zajęć**. To one są zawsze najbardziej przydatnym materiałem dydaktycznym dla nauczycieli. Są nie tylko źródłem pomysłów na to, jaką tematykę warto realizować, jakie stawiać cele i na jakie liczyć efekty, co i jak mają ćwiczyć uczniowie oraz jakimi narzędziami mają się posługiwać. Są także inspiracją do własnych opracowań. Zróżnicowane treści, proponowane do realizacji na kołach zainteresowań IT, wnoszą różnorodność i alternatywę dla standardowego materiału nauczania, umożliwiając fakultatywny wybór z dostosowaniem programu do potrzeb i możliwości uczniów. Konspekty-scenariusze zawierają powiązanie czynności uczniów z działaniami trenera i z mediami-środkami używanymi w danej fazie zajęć. Porządkują także chronologię czynności niezbędnych dla wykonania zadań.

Warto zwrócić uwagę, że Konspekty-scenariusze w proponowanej formie **nie są scenopisami** do wiernej realizacji krok po kroku. Zgodnie z nieodzownym stylem adaptacyjnym – tj. dynamicznym dopasowywaniem się do rzeczywistych sytuacji – i tak trener jest głównym animatorem tego, co dzieje się na zajęciach. W partycypacyjnym modelu także uczniowie stają się moderatorami narracji oraz współautorami dzieł, poprzez propozycje rozwiązań alternatywnych bądź rozszerzających funkcjonalność.

Konspekty są **zalecanymi wzorcami** do przemyślanego wykorzystania, przetestowanymi praktycznie i wyselekcjonowanymi jako wartościowy materiał dydaktyczny, który uzyskał wysoką ocenę uczestników testowania: uczniów, trenerów, nauczycieli-opiekunów obserwujących zajęcia oraz ekspertów. Rolą nauczycieli chcących wykorzystać te materiały jest trafny dobór modułów do możliwości uczniów i ułożenie ich odpowiedniej sekwencji w optymalny plan zajęć.

Zgodnie z nazwą projektu „Strategia wolnych i otwartych implementacji”, bardzo ważną rolę dydaktyczną pełnią tu **implementacje**. W tej Strategii edukacyjnej są to wszelkie wytwory w postaci cyfrowej i/lub elektronicznej, jakie powstają w toku rozwiązywania wyznaczonych zadań projektowych, programistycznych lub mechatronicznych. Implementacje są materializacją koncepcji algorytmicznych i urzeczywistnieniem modeli abstrakcyjnych. Sam proces przechodzenia od pomysłu do wytworu jest implementowaniem. Ze względu na to, że procesy inżynierskiego programowania czy konstruowania interfejsów są regulowane wypracowanymi specjalnie w tej dziedzinie metodykami postępowania (np. *programowanie zwinne*), tworzenie na kołach zainteresowań konkretnej implementacji samo w sobie ma niejako zawartą optymalną metodę dydaktyczną. Środowisko pracy i narzędzia służące do implementowania stają się wówczas środkami dydaktycznymi, a działający wytwór pełni dodatkowo funkcję poglądową. Zatem implementacje w Programie nauczania-uczenia się infotechniki są *integralnymi środkami-metodami*, w pełnym znaczeniu tego pojęcia.

Implementacje zalecane do wykorzystywania lub tworzone na zajęciach mają różną postać fizyczną. Najbardziej właściwą formą jest **postać cyfrowa**. W postaci opisu nie daje się odzwierciedlić w pełni istoty funkcjonowania takich wytworów. Ponadto nawet skrócone opisy implementacji powiązanych z Konspektami przekraczałyby dopuszczalną objętość publikacji. Z tego powodu w niniejszej publikacji zawarto jedynie opisy 11 wybranych z ponad stu przygotowanych i przetestowanych, natomiast wszystkie implementacje proponowane do realizacji na kołach zainteresowań IT umieszczono na płycie dołączonej do książki oraz w Serwisie e-Swoi.

W ogólnym ujęciu silnie zróżnicowanych form implementacji są to: opisy zadań, schematy układów, instrukcje postępowania lub przykładowe rozwiązania, łącznie z ilustracjami i kodami źródłowymi. Przyjęto założenie, że to trener w uzgodnieniu z uczniami podejmuje decyzję o tym, jaki fragment opisu implementacji udostępnia uczniom – czy tylko treść zadania do wykonania w całości od podstaw, czy część kodu źródłowego z lukami do wypełnienia, czy pełne kody z zadaniem modyfikacji lub rozszerzenia.

Indywidualne dopasowanie stopnia trudności zadań jest konieczne ze względu na silne zróżnicowanie potencjału uczniów. Z tego też powodu założone cele szczegółowe i oczekiwane efekty muszą być adekwatne do realnych możliwości. Wynika stąd zupełnie inny od tradycyjnego sposób oceniania osiągnięć. Podstawą jest **ocenie rzeczywiste**, polegające na obserwacji aktywności i czynności wykonywanych przez uczniów. Dotyczy to zarówno oceny jakości interakcji werbalnych, jak też prawidłowości rozwiązywania zadań i wykonania implementacji. Oprócz bezpośredniego wyrażania sobie opinii przez trenera, znamienna w proponowanym Programie jest możliwość wykorzystania przetestowanych na dużej próbie i wystandaryzowanych narzędzi do ukierunkowanej obserwacji, pogłębionej refleksji i do ewaluacji efektów.

Na potrzeby oceniania jakości realizacji kół zainteresowań IT udostępniamy *Arkusze obserwacji* przeznaczony dla osób hospituujących zajęcia. Także dla nich, lecz przede wszystkim dla trenerów polecamy *Protokół formatywny*, służący do refleksji odnoszonej do tego, jak w rzeczywistości udało się zrealizować najważniejsze elementy metodyczne, w tym innowacyjne. Te **narzędzia pomiarowe** są tak skonstruowane, że dotyczą obu podmiotów: trenera i uczniów, co umożliwia analizę oddziaływań i skutków. Do wartościowania efektów zajęć polecamy *Ankiety ewaluacyjną* dla uczniów, zbudowaną w formie skali psychometrycznej, służącej do pomiaru zmian, jakie zaszły pod wpływem całego cyklu zajęć w świadomości, postawach i cechach wolicjonalnych uczniów.

Oprócz dedykowanych narzędzi pomiarowych, nauczyciele otrzymują obszerny opis kompleksowej **metodologii** oceniania i ewaluacji, wraz ze schematami interpretacji jakościowej i analizy ilościowej w odniesieniu do norm wyznaczonych empirycznie dla populacji. Ta metodologia może być wzorcem dla opracowywania własnych narzędzi, przeznaczonych na inne zajęcia szkolne.

W realizacji Strategii edukacyjnej z obszarów infotechniki istotną rolę odgrywają **narzędzia pracy** uczniów i trenerów. Nie tylko dlatego, że od rodzaju wykorzystywanych systemów oprogramowania i aplikacji użytkowych zależy dostępność tych narzędzi w szkole, lecz także dlatego, że z założenia uczeń powinien mieć możliwość ćwiczenia także w domu lub w dowolnym innym miejscu. Przyjęto zasadę, że dedykowane do wykonywania implementacji narzędzia muszą być autonomiczne, niezależne od różnych systemów zainstalowanych na danym komputerze. Ponadto narzędzia te są udostępniane na liberalnych licencjach, aby ani szkoła, ani uczniowie nie ponosili kosztów zakupu systemu i oprogramowania. Wreszcie – niezbędny jest otwarty dostęp do kodów źródłowych i dokumentacji oprogramowania. Warunki te spełnia wykorzystanie **Wolnego i otwartego oprogramowania** (WiOO).

Założenia ideowe wolności i otwartości oraz ruchy społeczne wspierające te idee zrodziły się z potrzeby przeciwdziałania monopolizacji oprogramowania. Powstały i ciągle doskonalone są systemy oraz aplikacje oparte na wolnych i otwartych licencjach, zwykle zredukowanych jedynie do uznania autorstwa. Dzięki humanizacji informatyki i globalizacji dorobku ludzkości udostępniane są otwarte źródła, zasoby edukacyjne i naukowe, standardy technologiczne i metodologiczne, a także wolna kultura. Z tych dobrodziejstw korzysta Program nauczania-uczenia się infotechniki i obopólnie – dorobek Projektu SWOI jest udostępniany jako wolny i otwarty.

Platformą udostępniania dorobku i wymiany informacji jest **Serwis e-Swoi**. Skupia on Społeczność nazywaną „Swoi”, połączoną ideami nabywania umiejętności tworzenia implementacji i doskonalenia kompetencji infotechnicznych. Jego funkcjonalność wyraża się w kilku odmienionych usługach. Pierwszą z nich jest funkcja informacyjna, służąca otwartemu publikowaniu krótkich komunikatów, z mechanizmem promocyjnym *wyłączarni newsów*. Kolejnym miejscem publikacji jest *wiki*, gdzie umieszczane są dłuższe publikacje merytoryczne. Funkcjonalność o nazwie *program* ma na celu umożliwienie trenerowi budowę własnego planu zajęć poprzez wybór modułów. Poprzez usługę *e-tutor* użytkownik może skorzystać z doradztwa, zadać pytania i czytać odpowiedzi (FAQ) bądź dotrzeć do instrukcji. Struktura o skróconej nazwie *repo*, to obszar katalogów e-Repozytorium, w których umieszcza się i udostępnia własne implementacje i zasoby źródłowe. Funkcjonalność *e-portfolio* jest mechanizmem automatycznego rejestrowania aktywności publikacyjnej w Serwisie, a ponadto umożliwia pisanie osobistych refleksji o swym

udziale w Społeczności „Swoi”. W usłudze *konkurs* ogłaszane są zadania i warunki konkursowe, a następnie publikowane wyniki.

Oprócz platformy zdalnej, kompleksowym, dedykowanym narzędziem pracy własnej jest **Szkolny Remiks Ubuntu**, oparty na dystrybucji Linux Ubuntu ze środowiskiem graficznym XFCE. Cały pakiet oprogramowania został dobrany i skonfigurowany specjalnie do ćwiczeń i wykonywania implementacji według zadań z Konspektów-scenariuszy. Zawarto tam wszystkie niezbędne materiały dydaktyczne, pakiety edycyjne, środowiska programowania i multimedia.

Na potrzeby mobilności, system i aplikacje są uruchamiane bezpośrednio z pamięci Flash USB, więc nie ma potrzeby instalowania na twardym dysku. Na tym przenośnym nośniku są też zapisywane pliki użytkownika. Szkolny Remiks Ubuntu jest tak bogato wyposażony w sterowniki i aplikacje, że może być zainstalowany na osobistym komputerze lub notebooku jako w pełni wystarczalne środowisko pracy ucznia i nauczyciela, bez potrzeby zakupu żadnych innych systemów bądź programów.

Do zajęć z mechatroniki potrzebny jest zestaw zwany *modułem-interfejsem*. Funkcjonuje on we współpracy z komputerem poprzez złącze USB. Podstawowym elementem jest płytko **Arduino UNO R3** z mikrokontrolerem ATmega328. Drugim elementem jest uniwersalna płytka montażowa, służąca do łączenia podzespołów w układ elektroniczny. Do programowania zestawu służy interfejs IDE lub środowisko Scratch S4a. Są to środowiska interaktywne, pozwalające na sterowanie bądź odczyt wartości z czujników. W skład zestawu do zajęć wchodzi też podstawowe elementy elektroniczne: rezystory, tranzystor, termistor, diody LED i RGB, wyświetlacz LCD, czujniki, przycisk, buzzer itp. W zależności od projektu, wykorzystywane są różne podzespoły, tworząc alternatywne konfiguracje. Taki zestaw w przystępnej cenie może być znakomitym prezentem, sprawiającym podczas konstruowania niesamowitą radość – od najmłodszych, aż po dorosłych.

Jak już wspomnieliśmy – obszerny materiał, służący realizacji Programu nauczania-uczenia się infotechniki, został umieszczony w postaci elektronicznej na płycie dołączonej do wydania książkowego. Stanowi ona integralną część publikacji, jest zbiorem opisów, kodów implementacji i zasobów źródłowych, jest narzędziem pracy ucznia i trenera, nośnikiem systemu, aplikacji użytkowych i niezbędnego do zajęć oprogramowania. Ze względu na potrzebę aktualizacji, nowsze wersje będą udostępniane do pobrania z Serwisu pod adresem <http://e-swoi.pl/>

W tym miejscu chcemy podkreślić, że opis dorobku Programu „Strategia Wolnych i Otwartych Implementacji” zawarty jest w dwóch tomach. Oprócz niniejszego tomu pt. „Program nauczania-uczenia się infotechniki” integralną publikacją jest tom pt. „Strategia nauczania-uczenia się infotechniki”, która zawiera studium definicyjne i studium dydaktyczne. Teksty tam zawarte przybliżają założenia ideowe Strategii, taktykę jej wdrażania i realizacji oraz istotę proponowanych innowacji edukacyjnych. Gorąco zachęcamy Czytelników do zapoznania się z jednym i drugim tomem publikacji.

Stanisław Ubermanowicz, Krzysztof Wawrzyniak