

Program nauczania CodeInnova

Autorzy

Jukka Lehtoranta, Janne Fagerlund, Marika Peltonen, Denis Zhidkikh, Juho Liedes & Mikko Vesisenaho

Wydawca

Uniwersytet Jyväskylä / Projekt Erasmus+ CodeInnova

„Nauczanie programowania w szkole podstawowej: podstawa programowa, metody dydaktyczne, podręczniki, wsparcie online”

(numer projektu: 2019-1-PL01-KA201-065077)

Konsultanci / Recenzenci

Emilia Ahlström

Bogusława Denys

Dulce Freire

Katarzyna Garbacik

Bogusław Klimczuk

Maria José Rodrigues

Ivana Ružić

Vítor Silva

Marko Šolić

Siniša Stričak

Ana Trucek

Projekt i ilustracje

Jukka Lehtoranta

Martti Minkkinen

Tłumaczenie

Tanja Djaković (język chorwacki)

Maria José Rodrigues (język portugalski)

Bogusława Denys (język polski)

Projekt Erasmus+ CodeInnova

„Nauczanie programowania w szkole podstawowej: podstawa programowa, metody dydaktyczne, podręczniki, wsparcie online”

(numer projektu: 2019-1-PL01-KA201-065077)

Program nauczania CodeInnova

Struktura programu nauczania	3
Wiedza ogólna	3
Umiejętności myślenia komputacyjnego i programowania	3
Tworzenie treści cyfrowych	3
Wskazówki dla nauczyciela	3
Słowniczek	3
Podstawowe założenia myślenia komputacyjnego	4
Klasa 1	
Wiedza ogólna	5
Umiejętności myślenia komputacyjnego i programowania	5
Tworzenie treści cyfrowych	5
Wskazówki dla nauczyciela	5
Słowniczek	5
Klasa 2	
Wiedza ogólna	6
Umiejętności myślenia komputacyjnego i programowania	6
Tworzenie treści cyfrowych	6
Wskazówki dla nauczyciela	6
Słowniczek	6
Klasa 3	
Wiedza ogólna	7
Umiejętności myślenia komputacyjnego i programowania	7
Tworzenie treści cyfrowych	7
Wskazówki dla nauczyciela	7
Słowniczek	7
Klasa 4	
Wiedza ogólna	8
Umiejętności myślenia komputacyjnego i programowania	8
Tworzenie treści cyfrowych	8
Wskazówki dla nauczyciela	8
Słowniczek	8
Klasa 5	
Wiedza ogólna	9
Umiejętności myślenia komputacyjnego i programowania	9
Tworzenie treści cyfrowych	9
Wskazówki dla nauczyciela	9
Słowniczek	9

Klasa 6

Wiedza ogólna	10
Umiejętności myślenia komputacyjnego i programowania	10
Tworzenie treści cyfrowych	10
Wskazówki dla nauczyciela	10

Klasa 7

Wiedza ogólna	11
Umiejętności myślenia komputacyjnego i programowania	11
Tworzenie treści cyfrowych	11
Wskazówki dla nauczyciela	11
Słowniczek	11

Klasa 8

Wiedza ogólna	12
Umiejętności myślenia komputacyjnego i programowania	12
Tworzenie treści cyfrowych	12
Wskazówki dla nauczyciela	12
Słowniczek	12

Klasa 9 (8+)

Wiedza ogólna	13
Umiejętności myślenia komputacyjnego i programowania	13
Tworzenie treści cyfrowych	13
Wskazówki dla nauczyciela	13
Słowniczek	14

Źródła

Badania naukowe	14
Krajowe programy nauczania i podsumowania	15
Prognozy	15

Struktura programu nauczania

Program nauczania podzielony jest na bloki. Dodatkowo oferuje cele uczenia się, wspierające proces planowania (pod tytułem „Wskazówki dla nauczyciela”). Zawiera również słowniczek słów kluczowych.

Wiedza ogólna

Wiedza ogólna ma na celu zapewnienie narzędzi i umiejętności umożliwiających identyfikację programowania jako części rozwoju społecznego, rozwijanie myślenia wieloaspektowego oraz dostarczenie narzędzi, stanowiących integralną część procesu uczenia się przez całe życie.

Umiejętności myślenia komputacyjnego i programowania

Blok programowania wskazuje podstawowe umiejętności programowania (związane z podstawowymi zagadnieniami, praktyką i rozumieniem programowania), w tym również umiejętności myślenia komputacyjnego i abstrahowania na poziomie ogólnym.

Tworzenie treści cyfrowych

Tworzenie treści cyfrowych dostarcza tematy do programowania i przedstawia najważniejsze sposoby tworzenia treści cyfrowych i technologii, które będą najistotniejsze w nadchodzących dziesięcioleciach.

Wskazówki dla nauczyciela

Zawierają informacje dotyczące celów nauczania i planowania działań, w szczególności wspieranie myślenia i uczenia się, komunikacji i pracy grupowej oraz procesu nabywania umiejętności wieloaspektowych. Opisano również wskazówki dotyczące metod nauczania odpowiednich dla wieku uczniów. Nauczyciele mają dużą autonomię w korzystaniu z tego programu nauczania, mogą zmieniać kolejność treści i modyfikować poziom trudności.

Słowniczek

Ta sekcja zawiera definicje odpowiednich terminów.

Umiejętności wieloaspektowe Zestaw umiejętności związanych z myśleniem i komunikacją, które są potrzebne do interpretacji i tworzenia treści w różnych sytuacjach i środowiskach. Najistotniejszym elementem jest umiejętność pozyskiwania, modyfikowania, tworzenia, prezentowania, oceny i sortowania informacji.

Podstawowe założenia myślenia komputacyjnego

Myślenie komputacyjne jest procesem znajdowania rozwiązań do problemów różnych dziedzin przy świadomym wykorzystaniu metod i narzędzi informatycznych. Istnieje kilka zasad myślenia komputacyjnego. Obszary te zostały uwzględnione przy projektowaniu programu nauczania, ze szczególnym uwzględnieniem zasad myślenia programistycznego na poszczególnych poziomach nauczania.

Wspomnianie zasady to:

Abstrakcja Języki programowania, programy i dane są uogólnieniem zjawisk świata rzeczywistego

Algorytmy Skończona liczba kroków, instrukcji lub poleceń prowadzących do rozwiązania problemu

Automatyzacja Używanie zaprogramowanych instrukcji do rozwiązania zadań

Współpraca Wspólna praca i dzielenie się odpowiedzialnością

Kreatywność Tworzenie projektu jest zawsze formą twórczej ekspresji, programowanie wymaga znalezienia i wykorzystania różnych opcji.

Dane Korzystanie z różnorodnych danych, pochodzących z różnych źródeł

Liczby i uogólnienie liczb Powtórzenie ogólnych schematów rozwiązań do rozwiązywania podobnych problemów.

Iteracja Pierwotny pomysł jest ulepszany poprzez projektowanie, testowanie i korygowanie błędów, aż do osiągnięcia sytuacji idealnej.

Logika Programy zawierają różnorodne elementy logiczne, takie jak instrukcje warunkowe, logikę Boole'a i operacje arytmetyczne

Modelowanie i Projektowanie Programowanie obejmuje modele i algorytmy, które można później zaprogramować. Programowanie polega na dbaniu o strukturę, układ i funkcjonalność systemu.

Dekompozycja problemu Problemy można podzielić na mniejsze podproblemy, których rozwiązania są znane lub są łatwiejsze do rozwiązania

Wydajność Algorytmy nie zawierają zbędnych lub dodatkowych kroków

Rekoncylacja i podobieństwo Programy mogą wykonywać wiele operacji w tym samym czasie, co wymaga kontroli i zgrania w czasie

Testowanie i debugowanie Programiści śledzą kod, projektują i przeprowadzają testowanie oraz wychwytyją i naprawiają problemy.

Klasa 1

Wiedza ogólna

Uczniowie rozumieją znaczenie słów związanych z programowaniem, takich jak „programowanie” i „aplikacja”.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie uczą się rozwiązywania problemów krok po kroku i podstaw programowania w środowisku graficznym poprzez programowanie postaci i obiektów materialnych.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, iteracja, modelowanie i projektowanie, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie zapoznają się z obiektami materialnymi i programowaniem niecyfrowym.

Wskazówki dla nauczyciela

Uczniowie mają różne możliwości pracy, zarówno indywidualnej jak i grupowej.

Kwestie związane z omawianym tematem są podkreślane i poruszane w oparciu o codzienne doświadczenia uczniów.

Rozwój wieloaspektowości wspierany jest przez podejście multisensoryczne, holistyczne z uwzględnieniem świadomości fenomenalnej

Słowniczek

Graficzne środowisko programistyczne Programowanie opiera się na symbolach wizualnych w środowisku programowania graficznego, na przykład podawanie instrukcji ruchu za pomocą strzałek

Holistyczność Podejście kompleksowe

Programowanie niecyfrowe Programowanie za pomocą gier, symboli, instrukcji i innych środków niecyfrowych

Klasa 2

Wiedza ogólna

Uczniowie uczą się rozpoznawać otaczające ich kody i rozumieć cel programowania. Ponadto rozumieją, że każdy kod ma swojego twórcę i cel.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie odkrywają, projektują i tworzą kreatywne instrukcje krok po kroku, aby rozwiązać określone zadanie lub problem.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, kreatywność, dane, iteracja, logika, modelowanie i projektowanie, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie poznają programistyczne gry i animacje w graficznym środowisku programistycznym. Jednocześnie rozumieją potencjał programowania jako środka twórczej ekspresji.

Inne treści: przedmioty materialne, programowanie niecyfrowe

Wskazówki dla nauczyciela

Uczniowie zdobywają doświadczenie w zakresie różnych nawyków pracy związanych z programowaniem, korzystając z cyfrowych narzędzi komunikacji, wspierających naukę, komunikację i pracę zespołową.

Uczniowie są zachęceni do zadawania pytań, słuchania, dokonywania szczegółowych obserwacji, znajdowania informacji rozwijających już istniejące pomysły i wymyślania nowych, a także do przedstawiania swoich pomysłów.

Uczniowie są wspierani w rozwijaniu swoich umiejętności *wieloaspektowych* poprzez umożliwianie im interpretowania, tworzenia i rozwijania różnych kodów dostosowanych do ich wieku.

Słowniczek

Graficzne środowisko programistyczne Programowanie w tych środowiskach odbywa się w za pomocą bloków a nie poleceń tekstowych, na przykład w Scratch Junior (ScratchJr) lub Scratch.

Klasa 3

Wiedza ogólna

Uczniowie dowiadują się, jak decyzje człowieka wpływają na wydajność technologii.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie rozwiązują problemy, porządkują informacje i poznają pojęcia algorytmów.

Uczniowie uczą się wizualizować problemy za pomocą różnych wykresów i uogólnień

Uczniowie projektują, kodują i rozwijają programy za pomocą poleceń sekwencyjnych, selekcji i powtórzeń.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, automatyzacja, kreatywność, dane, iteracja, logika, modelowanie i projektowanie, problemy z partycjonowaniem, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie zapoznają się z zaawansowanymi urządzeniami mobilnymi oraz budują i programują roboty fizyczne lub wirtualne.

Inne treści: przedmioty materialne, programowanie niecyfrowe, proste gry i animacje

Wskazówki dla nauczyciela

Uczniowie zdobywają doświadczenie pracując w komfortowej grupie i tworząc wspólnie.

Uczniowie są wspierani w określeniu najbardziej dla nich odpowiednich sposobów uczenia się i rozwijają techniki uczenia się i innowacyjność.

Umiejętności wieloaspektowe promowane są poprzez analizę różnych kodów z perspektywy autora i użytkownika, z uwzględnieniem kontekstu i sytuacji.

Słowniczek

Algorytm Zestaw systematycznie wykonywanych instrukcji do rozwiązania określonego zadania

Polecenie Polecenie do sterowania urządzeniem

Powtórzenie Powtarzanie tej samej rzeczy kilka razy

Selekcja Wybór jednej lub kilku opcji

Robot wirtualny Symulator, który pozwala programować roboty bez urządzenia fizycznego.

Klasa 4

Wiedza ogólna

Uczniowie rozumieją potencjał programowania operacji automatycznych i symultanicznych

Umiejętności myślenia komputacyjnego i programowania

Uczniowie projektują i programują w graficznym środowisku programistycznym przy użyciu danych wejściowych. Uczniowie wprowadzają proste zmienne zawierające wartości liczbowe i teksty dla różnych obiektów.

Uczniowie potrafią rozwiązywać bardziej złożone problemy logiczne za pomocą technologii i bez niej. W rozwiązywaniu problemów wykorzystywane są instrukcje i zdarzenia krok po kroku oraz instrukcje warunkowe. Uczniowie uczą się korzystać z iteracji

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, współpraca, kreatywność, liczby i uogólnianie liczb, iteracja, logika, modelowanie i projektowanie, problemy z partycjonowaniem, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie poznają pojęcie sztucznej inteligencji i różne sposoby jej praktycznego wykorzystania.

Uczniowie tworzą gry na różnych platformach.

Inne treści: przedmioty materialne, programowanie niecyfrowe, proste gry i animacje, roboty fizyczne i wirtualne

Wskazówki dla nauczyciela

Uczniowie są wspierani w ocenianiu i rozwijaniu własnych umiejętności komunikacji i pracy zespołowej. Interaktywne uczenie się, zwłaszcza w grupie rówieśniczej, jest wykorzystywane na wiele sposobów, wzmacniając w ten sposób umiejętności pracy zespołowej uczniów.

Umiejętności myślenia są ćwiczone za pomocą rozwiązywania problemów i złożonych zadań oraz metod pracy, które wykorzystują i wspierają ciekawość, wyobraźnię, pomysłowość i uczenie się. Umiejętność programowania krytycznego rozwija się w kontekście kulturowym, odpowiednim dla uczniów i bliskim ich codziennym doświadczeniom.

Słowniczek

Sztuczna inteligencja Program zdolny do robienia tego, co uważa się za inteligentne

Wejście Np. numer wprowadzony w programie lub naciśnięcie przycisku, może być to również urządzenie pomocnicze podłączone do urządzenia, np. czujnik lub przycisk

Iteracyjne Używanie często występujących elementów tylko raz, tj. unikanie niepotrzebnego powtarzania kodu

Zmienna Miejsce przechowywania danych w programie

Klasa 5

Wiedza ogólna

Uczniowie zaczynają rozumieć, że kod jest zawsze matematycznym ćwiczeniem rozwiązującym dany problem oraz zauważają, że może to rodzić problemy etyczne.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie projektują i programują oprogramowanie, które wyświetla wartości zawierające liczby lub teksty. Uczniowie sprawdzają poprawność kodu, wykrywają i korygują błędy. Uczniowie tworzą proste zmienne. Uczniowie zapoznają się z przewidywaniem wyników, testowaniem i wyjaśnianiem istniejących programów.

Wyróżnione koncepcje myślenia obliczeniowego: Abstrakcja, algorytmy, automatyzacja, współpraca, kreatywność, liczby i uogólnianie liczb, logika, problemy z partycjonowaniem, wydajność, rekoncepcja i podobieństwo, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie projektują i tworzą prosty Internet Rzeczy i urządzenia ubieralne (wearables), a jednocześnie uczą się wyrażać siebie poprzez projektowanie i tworzenie.

Inne treści: przedmioty materialne, programowanie niecyfrowe, proste gry i animacje, roboty fizyczne i wirtualne, sztuczna inteligencja

Wskazówki dla nauczyciela

Zachęcamy uczniów do szukania sposobów ekspresji wspomagających współpracę.

Zachęcamy uczniów do poszukiwania kreatywnych rozwiązań. Uczniowie uczą się analizować różne cechy gier, programów i aplikacji, aby móc je rozróżniać.

Słowniczek

Internet rzeczy - Różne urządzenia podłączone do Internetu, np. czujniki pomiarowe lub sprzęt AGD

Wyjście — Wartość dostarczona przez program lub osprzęt podłączony do urządzenia

Wynik - Wartość, jaką wyświetla program, na przykład po operacjach matematycznych.

Urządzenia ubieralne- Na przykład smartwatch lub inteligentna odzież

Klasa 6

Wiedza ogólna

Uczniowie powinni omówić rolę programowania w aspekcie emocji. Uczniowie uczą się identyfikować wartości autora, które odzwierciedla kod.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie planują, przewidują, monitorują, tworzą i dostosowują programy odpowiednie do ich wieku. Uczeń będzie potrafił zakodować działający program i rozwiązywać bardziej złożone problemy, dzieląc je na mniejsze podproblemy. Uczniowie uczą się więcej o używaniu różnych rodzajów zmiennych i sami je tworzą.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, automatyzacja, współpraca, kreatywność, dane, liczby i uogólnianie liczb, iteracja, logika, modelowanie i projektowanie, problemy z partycjonowaniem, wydajność, rekuncylacja i podobieństwo, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie projektują i tworzą działającą grę lub program rozwiązujący konkretny problem (zmierzający do określonego celu).

Inne treści: przedmioty materialne, programowanie niecyfrowe, proste gry i animacje, roboty fizyczne i wirtualne, sztuczna inteligencja, Internet Rzeczy (IoT) i projektowanie urządzeń ubieralnych

Wskazówki dla nauczyciela

Uczniowie zbierają doświadczenia, aby jak najlepiej udostępnić grupie swoje umiejętności.

Prowadzenie obserwacji oraz korzystanie z różnorodnych źródeł informacji i narzędzi wzmacnia umiejętność zadawania pytań i znajdowania odpowiedzi, zarówno samodzielnie, jak i z innymi. Uczniowie proszeni są o porównanie i ocenę adekwatności kodu użytego do określonego celu.

Klasa 7

Wiedza ogólna

Uczniowie pogłębiają wiedzę na temat korzystania z różnego typu oprogramowania i związanych z tym zasad.

Umiejętności myślenia komputacyjnego i programowania

Uczeń będzie potrafił projektować i tworzyć programy wykorzystujące podprogramy, odpowiednie struktury i typy danych, wyrażenia, zmienne oraz polecenia iteracyjne i warunkowe. Ogólne języki programowania służą do tworzenia programów odpowiednich dla wieku. Uczeń rozumie różne sposoby wykorzystania symulacji i algorytmów sekwencyjnych do rozwiązywania problemów.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, automatyzacja, współpraca, kreatywność, dane, liczby i uogólnianie liczb, iteracja, logika, modelowanie i projektowanie, problemy z partycjonowaniem, wydajność, rekoncylacja i podobieństwo, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie tworzą bardziej złożoną grę, aplikację lub aplikację mobilną, która rozwiązuje konkretny problem z określonego tematu lub zakresu. Uczniowie uczą się, jak opisać działanie złożonego programu za pomocą wzorców i uogólnień.

Inne treści: gry i animacje, roboty fizyczne i wirtualne, sztuczna inteligencja, Internet Rzeczy (IoT) i projektowanie urządzenia ubieralne

Wskazówki dla nauczyciela

Uczniowie dostrzegają różne sposoby współpracy i różne role w grupie. Ponadto, zdobywają doświadczenie pracując w ramach tych ról.

Uczniowie są zachęceni do identyfikowania i ciągłego rozwijania własnych strategii programowania. Uczniowie rozwijają umiejętności analityczne, krytyczne i kulturowe związane z programowaniem.

Słowniczek

Typ danych - Pojęcie używane do definiowania zmiennych, które określają rodzaj zmiennej na przykład: string, integer

Ogólny język programowania - Programowanie odbywa się za pomocą języka programowania (na przykład Python lub Javascript)

Iteracja - Wyświetlanie tylko jednego powtórzenia, czyli unikanie niepotrzebnych powtórzeń w kodzie

Algorytm sortowania - Algorytm, który porządkuje listę w określonej kolejności w celu ułatwienia przetwarzania

Podprogram - Niezależna część programu, która wykonuje określoną funkcję

Klasa 8

Wiedza ogólna

Uczniowie pogłębiają zrozumienie znaczenia, potencjału i zagrożeń związanych z programowaniem na poziomie społecznym. Uczniowie uczą się posługiwać sztuczną inteligencją.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie potrafią projektować, tworzyć, dokumentować i prezentować programy i roboty, które rozwiązują konkretny problem z życia codziennego. Tworzone programy zawierają algorytmy wyszukiwania, tabele i funkcje automatyczne. W tych programach ma miejsce kilka jednoczesnych zdarzeń.

Wyróżnione koncepcje myślenia komputacyjnego: abstrakcja, algorytmy, automatyzacja, współpraca, kreatywność, dane, liczby i uogólnianie liczb, iteracja, logika, modelowanie i projektowanie, problemy z partycjonowaniem, wydajność, rekoncylacja i podobieństwo, testowanie i debugowanie

Tworzenie treści cyfrowych

Uczniowie tworzą bardziej złożone gry, aplikacje lub aplikacje mobilne, które symulują zagadnienia związane z życiem codziennym. Uczniowie poznają potencjał i możliwości bardziej zaawansowanych mikrokontrolerów.

Inne treści: gry i animacje, roboty fizyczne i wirtualne, sztuczna inteligencja, Internet Rzeczy (IoT) i projektowanie urządzeń ubieralnych

Wskazówki dla nauczyciela

Uczniowie znają i stosują różne metody komunikacji fizycznej i wirtualnej, współpracy i współdziałania w projektach programistycznych.

Uczniowie są zachęceni do wyrażania własnych doświadczeń we własnym sposobie myślenia, motywowani do słuchania siebie i innych oraz do patrzenia na zagadnienia z perspektywy innych. Zachęca się uczniów do wykorzystywania swoich umiejętności w zakresie umiejętności wieloaspektowych podczas aktywności w różnych sytuacjach

Słowniczek

Mikrokontroler System, który łączy analogowe komponenty elektroniczne (na przykład diody) z kodem zaprogramowanym przez użytkownika

Algorytm wyszukiwania Algorytm, który organizuje listę w określonej kolejności w celu ułatwienia przetwarzania

Klasa 9 (8+)

Wiedza ogólna

Uczniowie wykazują szerokie zrozumienie programowania, programów i ich roli we współczesnym społeczeństwie. Dodatkowo, rozumieją znaczenie programowania w wyrażaniu siebie.

Umiejętności myślenia komputacyjnego i programowania

Uczniowie projektują i wdrażają różne rozwiązania automatyzacyjne, a także analizują rozwiązania automatyzacyjne dla różnych aplikacji sprzętowych i programowych. Uczniowie będą badać możliwości rozwoju mobilnych systemów operacyjnych na praktycznych przykładach.

Wyróżnione koncepcje myślenia komputacyjnego: Abstrakcja, algorytmy, automatyzacja, kreatywność, liczby i uogólnianie liczb, iteracja, logika, problemy z partycjonowaniem, wydajność, rekondylacja i podobieństwo, testowanie i debugowanie

Zastosowane Zasady Myślenia Obliczeniowego: 1, 2, 3, 5, 7, 8, 9, 11, 12, 13, 14

Tworzenie treści cyfrowych

Uczniowie zapoznają się z technologią blockchain i jej zastosowaniami, rozumieją zasady działania rozwiązań symulujących łańcuchy bloków oraz zapoznają się z prostymi zasadami kryptograficznymi. Uczniowie zapoznają się z urządzeniami mobilnymi i zasadami działania ich systemów operacyjnych.

Inne treści: gry i animacje, roboty fizyczne i wirtualne, sztuczna inteligencja, Internet Rzeczy (IoT) i projektowanie urządzeń ubieralnych, mikrokontrolery

Wskazówki dla nauczyciela

Uczniowie będą dobrze posługiwać się różnymi pomocami programistycznymi (na przykład środowiskiem programistycznym) i różnymi metodami pracy. Uczniowie potrafią wyrazić swój własny sposób myślenia i obserwować sytuacje z perspektywy innych ludzi. Uczniowie posiadają umiejętność wykorzystywania własnych umiejętności wieloaspektowych, a także aktywnego i samodzielnego ich rozwijania.

Słowniczek

Technologia Blockchain Technologia umożliwiająca anonimowym podmiotom utrzymywanie rozproszonych baz danych, wykorzystywana np. w połączeniu z wirtualnymi pieniędzmi

Środowisko programistyczne Program lub zestaw programów służących do kodowania, środowisko może przetłumaczyć kod na program i automatycznie poprawić błędy w kodzie (na przykład naciśnięcia klawiszy)

Wirtualne pieniądze Cyfrowe pieniądze oparte na systemie szyfrowania, gdzie wszystkie transfery i własność są przechowywane w łańcuchu (blockchain)

Źródła

Badania Naukowe

Angeli C, Voogt J, Fluck A, Webb M, Cox M, Malyn-Smith J, Zagami J. A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. Educational Technology & Society. in 2016; 19 (3): 47-57

Barr V, Stephenson C. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads. 2011; 2 (1): 48-54

Czismadia A, Curzon P, Dorling M, Humphreys S, Ng T, Selby C, Woollard J. Computational thinking. A guide for teachers. 2015
<https://community.computingatschool.org.uk/resources/2324/single>

Fagerlund, J., Häkkinen, P., Vesisenaho, M. & Viiri, J. (2020). Computational thinking in programming with Scratch in Primary schools: systematic review. Computer Applications in Engineering Education, In press. DOI: 10.1002/cae.22255

Grover S, Pea R. Computational thinking: A competency whose time has come. In: Sentance S, Barendsen E, Schulte C, ed. Computer Science Education: Perspectives On Teaching And Learning In School. London: Bloomsbury Academic; 2018: 19-37.

Hsu TC, Chang SC, Hung YT. How to learn and how to teach Computational thinking: Suggestions based on a review of literature. Computers & Education. 2018; 126: 296-310.

Settle A, Perkovic L. Computational Thinking Across the Curriculum: A Conceptual Framework. 2010; Technical Reports, Paper 13
https://www.researchgate.net/publication/254582838_Computational_Thinking_across_the_Curriculum_A_Conceptual_Framework

Shute VJ, Sun C, Asbell-Clarke J. Demystifying Computational Thinking. Educational Research Review. 2017; 22: 142-158 Reports

Krajowe programy nauczania i podsumowania

Chorwacja: https://eacea.ec.europa.eu/national-policies/eurydice/content/croatia_en

Portugalia: https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal_en

Polska: https://eacea.ec.europa.eu/national-policies/eurydice/content/poland_en

Finlandia: https://eacea.ec.europa.eu/national-policies/eurydice/content/finland_enReports
oraz https://www.oph.fi/sites/default/files/documents/based_curriculum_country_2014.pdf

Prognozy

Adams Becker, S., Cummins, M., Freeman, A., and Rose, K. (2017). 2017 NMC Technology Outlook for Nordic Schools: A Horizon Project Regional Report. Austin, Texas: The New Media Consortium. <https://eric.ed.gov/?id=ED593946> Educause horizon report, 2019
<https://library.educause.edu/resources/2019/4/2019-horizon-report>