

# Program nauczania CodeInnova

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

## Konsultanci / Recenzenci

Emilia Ahlström  
Bogusława Denys  
Dulce Freire  
Katarzyna Garbacz  
Bogusław Klimczuk  
Maria José Rodrigues  
Ivana Ružić  
Vítor Silva  
Marko Šolić  
Siniša Stričak  
Ana Trucek

## Tłumaczenie

Tanja Djaković (j.chorwacki)  
Maria José Rodrigues (j.portugalski)  
Bogusława Denys (j.polski)

## 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)

## Projekt i Ilustracje

Jukka Lehtoranta  
Martti Minkkinen

## Program nauczania CodeInnova

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.

Wiedza

### Umiejętności myślenia komputacyjnego i programowania

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

Umiejętności

### 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.

Umiejętności praktyczne

Inne treści:  
Podsumowanie zagadnień jako punkt wyjścia do tworzenia treści cyfrowych

## 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.

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.

### Project Erasmus+ CodeInnova

Materiał opracowany przez Jukka Lehtoranta, Wydział Edukacji Uniwersytetu Jyväskylä (jukka.pe.lehtoranta@jyu.fi)

# Klasa 1

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.

## 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.

**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

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.

## 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.

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

# Klasa 3

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.

## 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.

### 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

Pupils understand the potential of programming for automatic and simultaneous operations.

## 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

## 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.

### 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

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.

## 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.

### 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

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.

## 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

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.

## 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.

### 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

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.

## 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

**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+)

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.

## 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.

### 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)



# Podstawowe pojęcia myślenia komputacyjnego

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 - Powtarzanie 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

Rekoncyliacja 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.

# 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](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](https://eacea.ec.europa.eu/national-policies/eurydice/content/croatia_en)

**Portugalia:** [https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal_en)

**Polska:** [https://eacea.ec.europa.eu/national-policies/eurydice/content/poland\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/poland_en)

**Finlandia:** [https://eacea.ec.europa.eu/national-policies/eurydice/content/finland\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/finland_en) Reports oraz

[https://www.opf.fi/sites/default/files/documents/based\\_curriculum\\_country\\_2014.pdf](https://www.opf.fi/sites/default/files/documents/based_curriculum_country_2014.pdf)

