

# Programowanie w szkole zgodnie z nową podstawą programową

---

Damian Kurpiewski

ODN Poznań, 25.10.2017

# Programowanie – dlaczego jest trudne?

---

- Składnia języka do zapamiętania
- Niewielki błąd - wielka katastrofa
- Częste testowanie
- Mało atrakcyjna forma i wyniki
- Dużo pracy i mierny wynik
- Abstrakcyjna forma rozumowania
- Komputer - często większa przeszkoda niż pomoc



# Programowanie (algorytmika) – jak uczyć?

---

- Na naukę jednego zagadnienia warto przeznaczyć kilka godzin
- Komputery rozpraszają – uczniowie powinni do nich siadać dopiero po zrozumieniu tematu
- Warto formułować praktyczne problemy, których rozwiązania poszukujemy

# Programowanie (algorytmika) – schemat lekcji

---

1. Zapoznanie z tematem – opis teoretyczny
2. Postawienie problemu
3. Rozwiązanie problemu „na sucho” – algorytm
4. Implementacja rozwiązania
5. Sprawdzenie wiedzy na podstawie podobnego (ale innego) problemu

# Programowanie w nowej podstawie

---

- Klasy I – VI: programowanie wizualne
- Klasy VII – VIII: programowanie tekstowe



# Programowanie wizualne

---

# Programowanie wizualne - narzędzia

Scratch

Blockly

Baltie

Logo

# Programowanie blokowe (puzzle)

---

- Zamiast kodu - bloki (puzzle)
- Wszystkie instrukcje dostępne w jednym miejscu
- Składamy bloki - tworzymy program
- Wiadomo, co do siebie pasuje
- Często ograniczone do konkretnego zastosowania



# Scratch

---

- Przeznaczenie - klasy 1-3 (może 4-5)
- Tworzenie interaktywnych historii i gier
- Sterowanie duszkami (sprites)
- Ciężki do nauki algorytmów
- Mało atrakcyjny dla starszych
- Adres: <https://scratch.mit.edu/>

# Scratch przykład

---



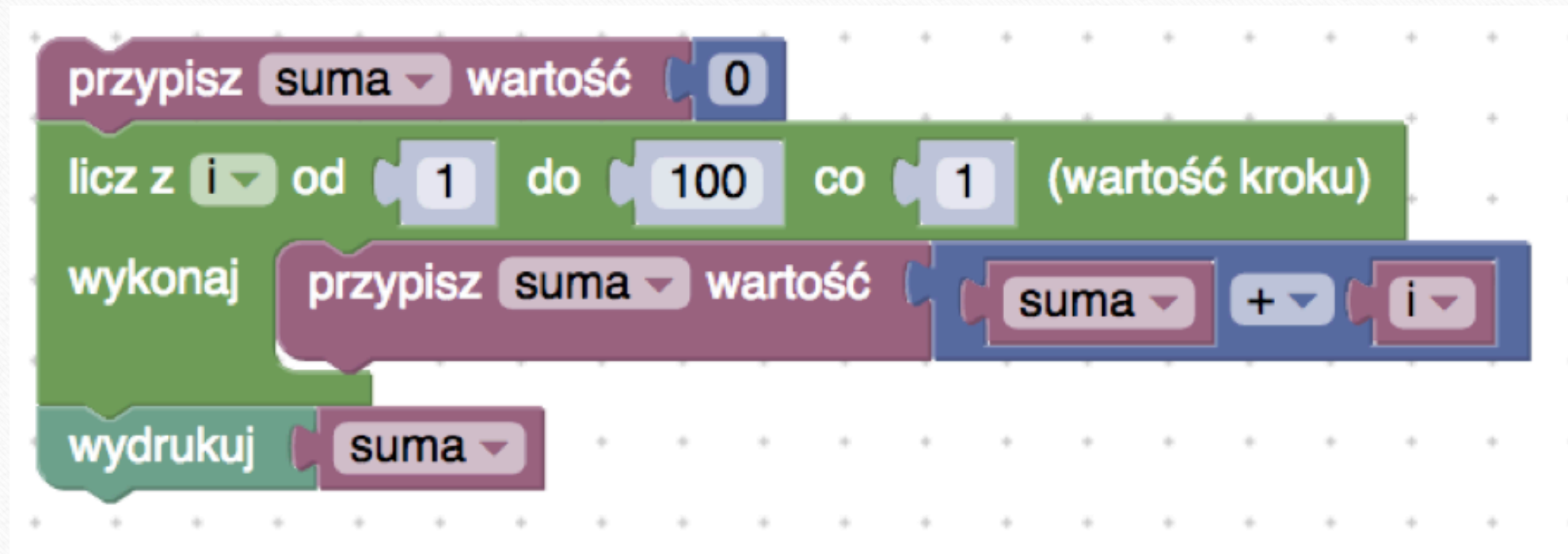


# Blockly

---

- Kod programu przedstawiony w postaci bloczków
- Atrakcyjny, graficzny interfejs i czytelny podział bloków instrukcji
- Zestaw gotowych, ciekawych gier i problemów
- Możliwości języka JavaScript
- Powszechnie stosowany

# Kod w postaci bloków





# Blockly – wykorzystanie na zajęciach

---

- Blockly Games – gotowy zestaw ćwiczeń o różnych stopniach zaawansowania
- Idealne do zastosowania zarówno na lekcji, jak i do samodzielnej pracy uczniów
- Możliwość zapisania rozwiązania w postaci linku
- Stopniowe wprowadzanie nowych instrukcji (bloków)
- Stopniowo rosnący poziom trudności

# Blockly Games

---

<https://blockly-games.appspot.com/>



# Przykłady lekcji

---

# Wprowadzenie do algorytmów

---

1. Zapoznanie się z ideą algorytmiki: przykłady algorytmów, instrukcje warunkowe, pętle
2. Blockly Labirynt: rozwiązywanie zadań 1-9
3. Znalezienie wyjścia z labiryntu – algorytm lewej ręki – konstrukcja i implementacja w Blockly Labirynt – zadanie 10
4. Lekcja ewaluacyjna – zmodyfikowane zadanie z labiryntem – wymagana implementacja algorytmu prawej ręki



# Wprowadzenie do algorytmów

---

Prezentacja materiałów

# Wprowadzenie do rekurencji

---

1. Zapoznanie z istotą rekurencji – zadanie z przeszukiwaniem pudełek
2. Fraktale – informacje ogólne; konstrukcja Trójkąta Sierpińskiego – algorytm
3. Trójkąt Sierpińskiego – implementacja w Blockly Żółw
4. Lekcja ewaluacyjna – rekurencyjne przechodzenie labiryntu – na przykładzie labiryntu gdzie metoda prawej/lewej ręki nie działa



# Wprowadzenie do rekurencji

---

Prezentacja materiałów

# Blockly Code

---

- Szersze zastosowanie – możliwości języka programowania...
- jednak częściowo ograniczone
- Możliwość implementacji algorytmów i interaktywnych programów
- Możliwość podejrzenia implementacji w różnych językach skryptowych: JavaScript, Python, ...



# Blockly Code – test pierwszości

---

[https://blockly-  
demo.appspot.com/static/demos/code/index.html?lang=pl#izofsd](https://blockly-demo.appspot.com/static/demos/code/index.html?lang=pl#izofsd)

# Blockly Code – sito Erastostenesa

---

[https://blockly-  
demo.appspot.com/static/demos/code/index.html?lang=pl#3mjh9c](https://blockly-demo.appspot.com/static/demos/code/index.html?lang=pl#3mjh9c)

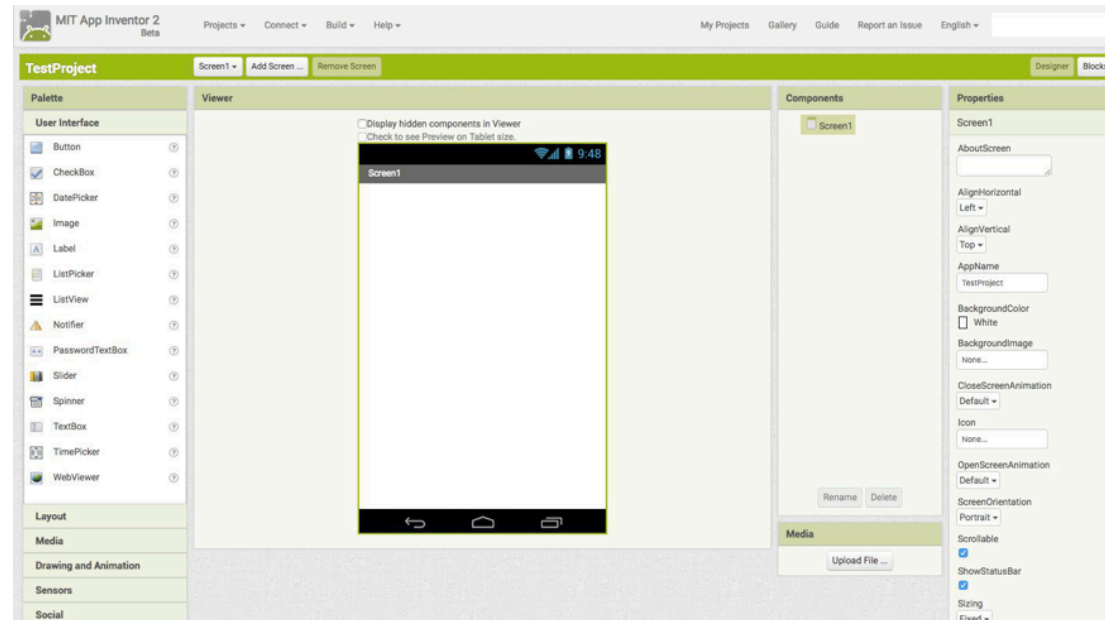


# Blockly - zastosowania

---

# MIT App Inventor 2

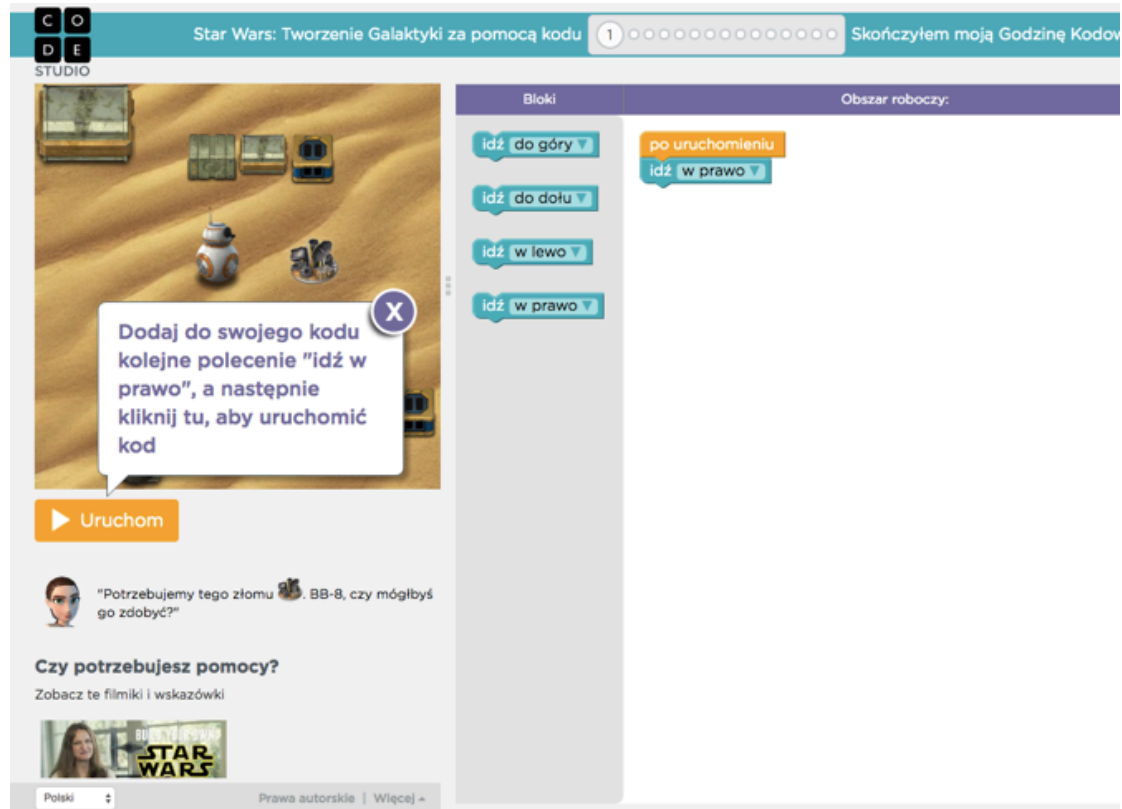
<http://ai2.appinventor.mit.edu/>





# Godzina kodowania

---





Bring  
coding to  
life.

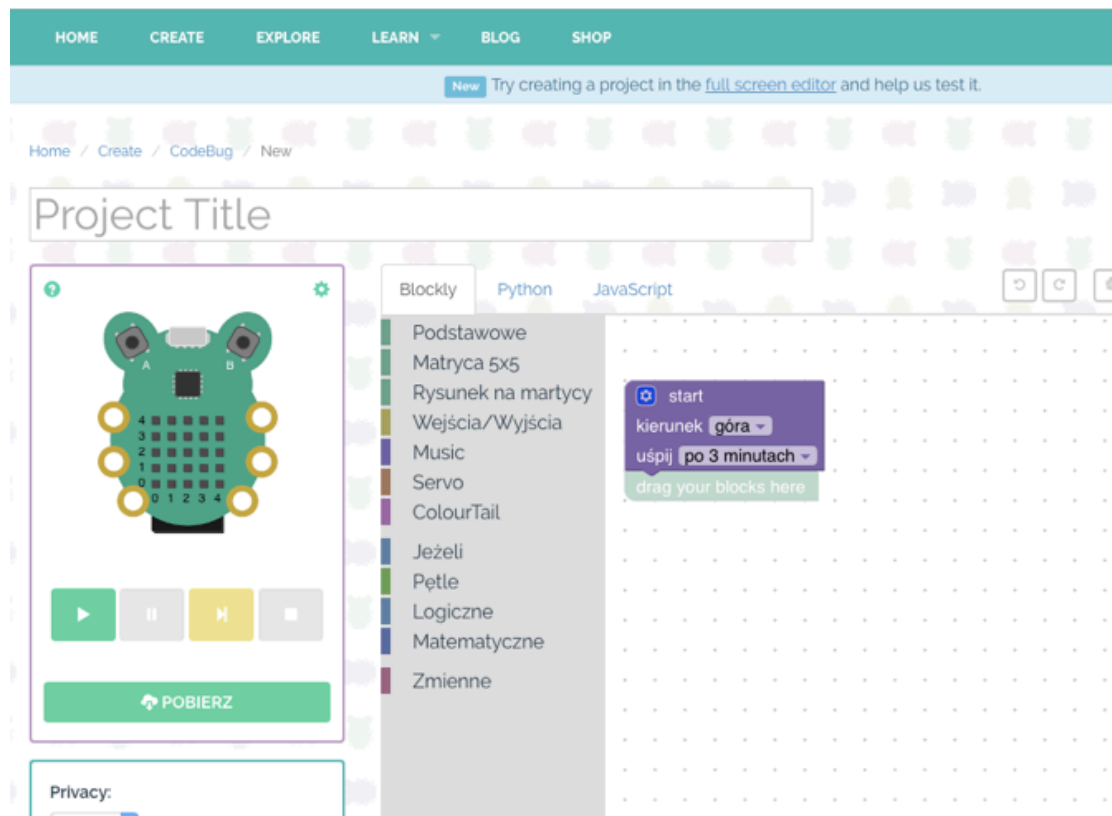
# OzoBlockly

---



# CodeBug

---







# Code Spells

---

Build elegant IoT applications with javascript simplicity and the agile visual development

script;



# Baltie

---

- Z wierzchu – język graficzny
- Od spodu – C#
- Różnorodne tryby pracy – od prostych po bardziej zaawansowane





# Programowanie tekstowe

---

# Programowanie tekstowe - klasyka

Pascal

C++

Python

Java



# Programowanie tekstowe - rozszerzenie

---

Ruby  
SonicPi

Lua  
Pico-8

JavaScript

Processing

# Pascal - plusy

---

- Język typowo dydaktyczny
- Stworzony do nauki programowania
- Wymusza zachowanie struktury kodu
- Uczy pisanie przejrzystego i czytelnego kodu

# Pascal - minusy

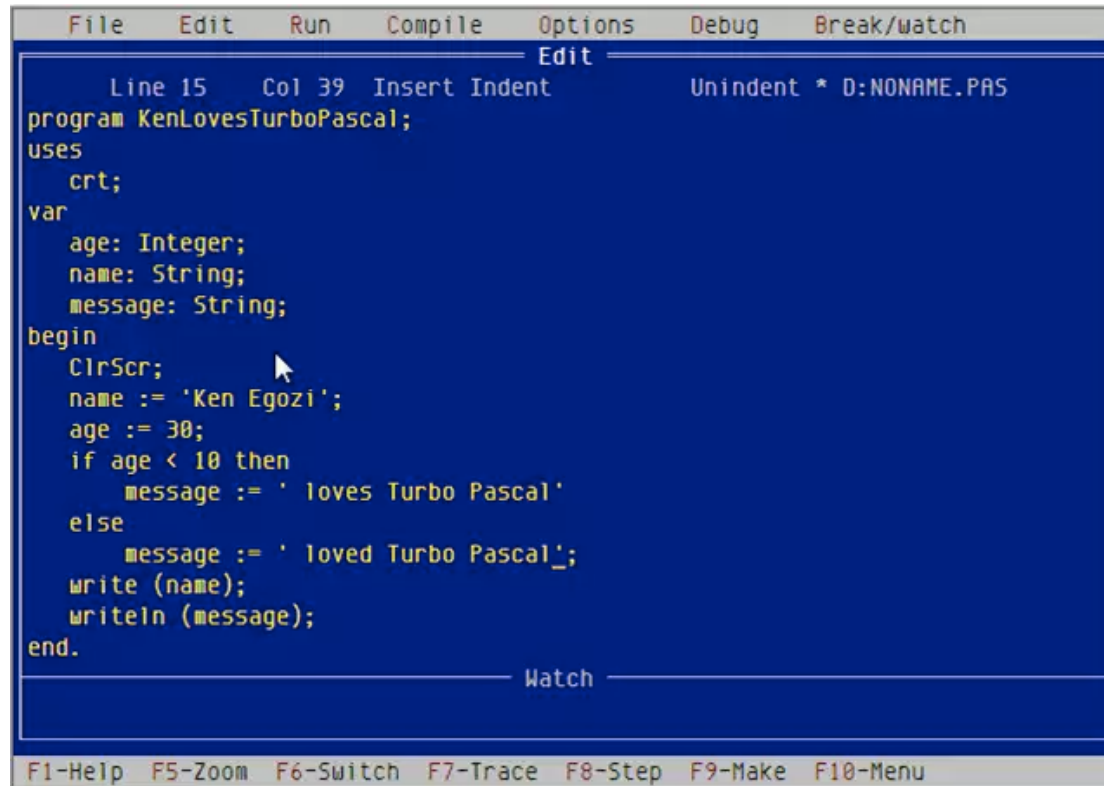
---

- Język przestarzały – brak praktycznych zastosowań
- Pod względem kariery zawodowej – nieprzydatny
- Toporne środowisko programistyczne (IDE)



# Pascal przykład

---



The screenshot shows the Turbo Pascal IDE interface. The menu bar at the top includes File, Edit, Run, Compile, Options, Debug, and Break/watch. Below the menu bar, the status bar indicates 'Line 15 Col 39 Insert Indent' and 'Unindent \* D:NONAME.PAS'. The main editing area contains the following Pascal code:

```
program KenLovesTurboPascal;  
uses  
  crt;  
var  
  age: Integer;  
  name: String;  
  message: String;  
begin  
  ClrScr;  
  name := 'Ken Egozi';  
  age := 30;  
  if age < 10 then  
    message := ' loves Turbo Pascal'  
  else  
    message := ' loved Turbo Pascal';  
  write (name);  
  writeln (message);  
end.
```

Below the code editor is a 'Watch' window, which is currently empty. At the bottom of the IDE, a status bar displays function key shortcuts: F1-Help, F5-Zoom, F6-Switch, F7-Trace, F8-Step, F9-Make, and F10-Menu.

# Python - plusy

---

- Język skryptowy
- Niski próg wejścia
- Brak „kontroli typów”
- Brak sztywnej struktury programu
- Ogromna ilość bibliotek
- Szeroki zakres zastosowań – algorytmika, automatyzacja zadań, pobieranie danych z Internetu, tworzenie wykresów, animacje, aplikacje okienkowe, gry, obliczenia matematyczne, serwisy webowe...



# Python - minusy

---

- Większa automatyzacja – „język” robi więcej za nas
- Łatwiej napisać „błędny”, ale działający program
- Nauczyciel musi położyć większy nacisk na poprawne konstrukcje
- Gdy potrzebujemy prędkości i większej kontroli – czas na zmianę języka



# Python - Uwaga

---

- Należy zdecydować się na jedną z wersji języka: 2 lub 3
- Pomiedzy wersjami są drobne (ale też i większe) różnice, np. nawiasy przy funkcji print

# Python - IDE

---

- Pycharm Community Edition

<https://www.jetbrains.com/pycharm/?fromMenu>

- Pycharm Edu

<https://www.jetbrains.com/pycharm-edu/?fromMenu>

# Python - turtle

---

- Sterowanie żółwiem
- Alternatywa dla klasycznego LOGO
- Proste użycie - wystarczy zaimportować bibliotekę
- Intuicyjne polecenia: forward, left, right...



# Python - turtle

---

Demo

# Python - algorytmy

---

<https://github.com/blackbat13/Algorithms-Python>

# C++ - cechy

---

- Kontrola typów
- Kompilacja kodu – wykrywanie błędów składni
- Ustalona struktura programu



# C++ - plusy

---

- Język znany przez praktycznie każdego programistę
- Większy narzut na programistę – trudniej napisać „błędny”, ale działający program
- Wiele języków programowania jest podobnych do C++
- Wykorzystywany przy „dużych” projektach – silniki graficzne, aplikacje okienkowe...
- Prędkość działania programów
- Wykorzystywany na zawodach programistycznych, np. Olimpiada Informatyczna

# C++ - minusy

---

- Wiele informacji do zapamiętania – struktura programu, biblioteki, typy zmiennych...
- Łatwo napisać dobry, poprawny, ale „brzydki” program
- Dla początkujących programistów: ograniczenie do aplikacji konsolowych
- Ciężkie rozszerzenie do innych zastosowań
- Dla osób niezbyt zainteresowanych programowaniem: trudny i nudny

# Java - cechy

---

- Język obiektowy
- Kontrola typów
- Ustalona struktura programu
- Łatwy sposób dołączania kolejnych plików



# Java - plusy

---

- Prostsze tworzenie aplikacji okienkowych
- Bardzo popularna na rynku pracy
- Możemy w niej pisać aplikacje na smartfony – system Android

# Java - minusy

---

- Jeszcze więcej informacji do przyswojenia
- Zrozumienie idei programowania obiektowego
- Do nauki algorytmów – jak strzelanie z kuszy do muchy
- Ciężka, toporna i wolna w działaniu

# Java - IDE

---

- IntelliJ Idea Community Edition  
<https://www.jetbrains.com/idea/>



# Pascal vs C++ vs Python vs Java

---

- Pascal jest dobry – ale jak uzasadnić jego użycie uczniom?
- Ma ograniczone zastosowanie
- C++ – do nauki algorytmów wręcz idealny...
- chyba że wolimy poświęcić się zrozumieniu idei
- Może być nudny dla „nie-programistów”

# Pascal vs C++ vs Python vs Java

---

- Python – dobry do rozpoczęcia przygody z programowaniem
- Początkowo nie wymaga przyswojenia wielu informacji
- Java – bardzo dobra gdy chcemy tworzyć zaawansowane aplikacje

# Języki alternatywne

---

Czyli co w trawie piszczy



# Ruby - SonicPi

---

- Programowanie w języku Ruby - języku skrypcowym
- Tworzenie muzyki w łatwy sposób
- Dobry do nauki i zrozumienia pętli
- Zintegrowany tutorial – wiele gotowych ćwiczeń i przykładów
- Adres: <http://sonic-pi.net/>

# SonicPi

---

Demo

# Lua – Pico-8

---

- Lua – język skryptowy, podobny do języka Python
- Pico-8 – wirtualna, 8-bitowa konsola
- Zintegrowane środowisko programistyczne – uruchamianie programów, edytor kodu, edytor duszków i map, edytor efektów dźwiękowych i muzyki
- 8-bitowe środowisko: ekran 128x128 pikseli, paleta 16 kolorów
- „Żywy organizm” – zmieniamy fragment kodu, uruchamiamy i sprawdzamy – bez dodatkowego czekania



# Pico-8

---

- Wsparcie do tworzenia animacji – w formacie gif
- Prosta i szybka publikacja stworzonych programów
- Wszystkie programy są otwarto-źródłowe: każdy można pobrać, zobaczyć, zmodyfikować, uczyć się
- Aktywna i przyjazna społeczność
- Adres: <https://www.lexaloffle.com/pico-8.php>

# Pico-8

---

Demo

# Pico-8 - sortowanie

---

Demo



# Pico-8 - inspiracje

---

<https://github.com/felipebueno/awesome-PICO-8>

# JavaScript

---

- Język Internetu - technologia powszechnie używana
- Język skryptowy – podobny do języka Python
- Może być stosowany do urozmaicenia stron...
- jak i do tworzenia samodzielnych aplikacji

# JavaScript – matematyczne animacje

---

Demo



# JavaScript - labirynt

---

Demo

# Processing

---

- Stworzony z myślą o artystach
- Duże możliwości tworzenia interaktywnych programów i animacji
- Wiele gotowych i ciekawych przykładów
- Adres: <https://processing.org/>

# Processing

---

Demo



# Zadania – skąd czerpać inspiracje?

---

I jak aktywizować uczniów

# Serwisy z zadaniami

---

- SPOJ - polski i angielski
- UVA Online Judge
- CodeInGame
- CodeCademy

# Polski SPOJ

---

- Zadania w języku Polskim
- Pogrupowane pod względem trudności
- Aktywne Polskie forum
- Komentarze do zadań
- Wiele dostępnych języków
- Łatwo się wybić



# UVA

---

- Zadania w języku angielskim
- Światowy serwis i ranking
- Ponad 5 tysięcy zadań
- Aktywna i rozległa społeczność
- Adres: <https://uva.onlinejudge.org/index.php>

# UVA - uHunt

---

- Podgląd profilu - status wysłanych rozwiązań, zadania rozwiązane i próbowane
- Wybrane zadania podzielone na 9 kategorii i liczne podkategorie
- Książka z opisem każdej kategorii, sposobów rozwiązywania, algorytmów i nie tylko
- Lista kolejnych zadań do rozwiązania
- Adres: <http://uhunt.onlinejudge.org/>

# UVA - uDebug

---

- Sprawdzenie, jakiej odpowiedzi powinien udzielić program
- Przykładowe przypadki testowe
- Możliwość porównania wyników
- Szybki sposób na znalezienie błędu



# CodeInGame

---

- Praktyczne problemy symulowane za pomocą gry
- Zadania algorytmiczne
- Możliwość zobaczenia, jak radzi sobie nasze rozwiązanie
- Zadania o różnym poziomie trudności
- Częste konkursy i zawody
- Wiele języków do wyboru
- Adres: <https://www.codingame.com/>

# CodeCademy

---

- Nauka programowania (i nie tylko) w ciekawy sposób
- Gotowe, interaktywne kursy: język Python, Ruby, tworzenie stron internetowych...
- Wyłącznie w języku angielskim
- Jako inspiracja dla nauczyciela...
- i do samodzielnej nauki dla ucznia
- <https://www.codecademy.com/>

# MWPZ

---

- Mistrzostwa Wielkopolski w Programowaniu Zespołowym
- Odbývają się regularnie w grudniu w Poznaniu
- Jedyne tego typu zawody otwarte dla wszystkich
- Dobra okazja do nauki pracy w zespole



Dziękuję za uwagę

---

I zapraszam do zadawania pytań

# Źródła

---

- [http://blog.mikaellundin.name/assets/posts/2011-02-14-beginners-guide-to-pascal/turbo\\_pascal.png](http://blog.mikaellundin.name/assets/posts/2011-02-14-beginners-guide-to-pascal/turbo_pascal.png)
- [http://progopedia.com/static/upload\\_img/2011/03/15/baltie3-fact.png](http://progopedia.com/static/upload_img/2011/03/15/baltie3-fact.png)