#### PROGRAMOWANIE PODRĘCZNIK DLA NAJMŁODSZYCH DZIECI

#### autor: Jerzy Wawro

Jarosław, 2017

wydanie I (poprawione - wersja 1.2)

#### Dodatkowe informacje o podręczniku:

ISBN 978-83-63384-08-1

Skład i oprawa graficzna: Jerzy Wawro Copyright (C) Fundacja Galicea

Fundacja Galicea udziela zezwolenia na wykorzystanie wersji elektronicznej podręcznika w nauczaniu w polskich szkołach. Zezwolenie obejmuje prawo do powielania (także w formie druku) w celach związanych z nauczaniem. Prawo to nie obejmuje możliwości wykorzystania komercyjnego oraz kopiowania fragmentów dzieła w celu wykorzystania go w innych dziełach (poza cytowaniem dozwolonym przez prawo). Użyte w podręczniku grafiki są dziełem autora, pochodzą z serwisu freepik.com (oznaczone zgodnie z licencją) lub zostały zakupione w serwisie fotolia.com. Prawo cytowania nie obejmuje tych ilustracji.

Osoby zainteresowane wersją drukowaną podręcznika, prosimy o kontakt z Fundacją: fundacja@galicea.org

Strona z programami i materiałami pomocniczymi do podręcznika: http://otwartaedukacja.pl/programowanie

Temat 1. Programiści mimo woli

# **PROGRAM = PLAN DZIAŁANIA**

Człowiek wpływa na otoczenie. Może to robić **metodą prób i błędów**, albo **planując działania** na podstawie znajomości praw przyrody

i konstrukcji urządzeń.

Przykład (nie rób tego w domu) - plan powodzi w mieszkaniu: **1) zatkaj zlew;** 

2) odkręć kurek i odpowiednio długo poczekaj ;-).

# Maszyny i urządzenia

Dla zilustrowania planowanego wpływania na działanie urządzenia (programowania) możemy zbudować własną prostą maszynę (na przykład korzystając z pochylonego stołu):

> Przestawiając fioletowy klocek na prawą lub lewą stronę – decydujemy o tym do którego pudełka spadnie piłeczka.



# Gry

Istnieje kilka gier rozwijających umiejętność programowania rozumianego jako **plan działania**. Na przykład tory dla kulek ("Marble") z plastiku lub drewna:



Tor składa się z elementów – rurek i "zwrotnic" - tak aby kulka puszczona z góry trafiła w określone miejsce.

Do tego typu gier należą także niektóre gry komputerowe.

Odwiedź stronę:

http://otwartaedukacja.pl/programowanie

#### Cele lekcji

1. Zrozumienie określeń: metoda prób i błędów, prawa przyrody, konstrukcja urządzeń, plan działania.

2. Rozwój umiejętności planowania. Układanie planów jako programowanie.

# **PROGRAMOWANIE AUTOMATÓW**

# Stan urządzenia

Jednym z najważniejszych słów w informatyce (i technice) jest słowo "**stan**". Stosujemy go w podobnym znaczeniu, jak w języku potocznym (używanym w rozmowach). Mówimy, że samochód jest w dobrym stanie, albo stan drogi jest "opłakany". W działaniu urządzeń "stan" to inaczej położenie, etap działania (określenie co teraz dzieje się z urządzeniem).

#### Przykłady:

- odkurzacz może być w stanach: włączony lub wyłączony
- światła dla pieszych mogą być w stanach: świeci się zielone, świeci się
- czerwone światła dla pieszych (stan = jaki kolor włączony)

Określenie "stan" zwraca uwagę na to co w działaniu urządzenia jest najważniejsze. Opis działania urządzeń może sprowadzać się do opisów tego w jaki sposób urządzenie przechodzi od jednego stanu do drugiego.

# Automat

Automat to urządzenie zmieniające stan bez ingerencji człowieka. Człowiek może zainicjować proces zmiany stanu. Później zmiany następują automatycznie.

#### Przykłady:

• światła dla pieszych (po naciśnięciu przycisku następuje przechodzenie przez kolejne stany)

• winda (po wybraniu piętra wykona po kolei: zamknięcie drzwi, uruchomienie silnika, zatrzymanie, otwarcie drzwi na określony czas i wreszcie ponowne zamknięcie drzwi).

#### **Pytania:**

• Przez jakie stany przechodzą światła na przejściu dla pieszych po ich włączeniu guzikiem?

• Podaj inne przykłady automatów (automat parzący kawę, pralka automatyczna).



### Program

Program to szczegółowy plan działania.

#### Przykłady programów z życia wziętych:

1. Program wycieczki (przykład stanu: jeśli jest godzina 12:00 i jesteśmy na

Krakowskim rynku, to słuchamy hymnu z Wieży Mariackiej).

2. Program telewizyjny.

W przypadku automatów **program definiuje** (określa, opisuje) kolejne stany przez jakie przechodzi automat (taki jak winda, światła na przejściu dla pieszych lub pralka automatyczna). Uruchamiając pralkę mówimy o wybraniu programu, albo uruchomieniu programu. **Program to opis kolejnych stanów i sposobu ich zmiany**. Znając stan aktualny i wykonywany program, możemy ustalić stan następny. Czasem program ma kilka **ustawień (wariantów**) – na przykład pralka może prać w różnych temperaturach.



#### Cele lekcji

1. Zrozumienie określeń: automat, automatyczny, stan, ustawienia, program.

2. Układanie programów jako wybór stanów przez jakie przechodzi automat (ustawienia).

#### Podsumowanie tematu

1. Programowanie to przygotowania dla automatów instrukcji "samodzielnego" działania.

2. Umiejętność programowania różnych urządzeń jest powszechna – pomimo, że brakuje umiejętności bardziej świadomego (a więc i bardziej twórczego) działania w tym zakresie.

3. Program to informacja o oczekiwanych zmianach stanu. Automat – zmienia stan zgodnie z programem (bez ciągłej ingerencji człowieka).

## ABSTRAKCJA

**Zadanie**: Piszemy instrukcję ustawiania budzika (może być w smartfonie). Po zakończeniu opisu chowamy budzik i próbujemy odpowiedzieć na pytania:

- Czy opis jest zrozumiały mimo że nie widzimy opisywanego przedmiotu?
- A gdy ten przedmiot zupełnie ulegnie zniszczeniu?

Bierzemy inny budzik i sprawdzamy, że instrukcja wciąż działa. Nasz opis nie odnosi się do konkretnego budzika, ale do każdego podobnego budzika. Inaczej mówiąc – do **abstrakcyjnego** wyobrażenia budzika.

Obserwowany cień konkretnego drzewa nie jest abstrakcją. Gdy jednak mówimy o powszechnie obserwowanych w słoneczne dni cieniach – posługujemy się abstrakcją.

### Symbole

Abstrakcyjny opis jest łatwiejszy, gdy używa się ilustracji. Dzięki nim nasze osobiste wyobrażenie można utrwalać i przekazywać. Wystarczy uproszczony rysunek budzika, by wiedzieć o czym rozmawiamy.



Taki rysunek to nie tylko artystyczne przedstawienie rzeczy. Każdy może sobie skojarzyć go z czymś co widział – konkretnym urządzeniem.

# Jeśli informacje przekazywane w taki sposób są powszechnie akceptowane (tak jak uśmiechnięte słońce) – to takie rysunki nazywamy symbolami.

#### Pytania pomocnicze:

- Jakie jeszcze znamy symbole?
- Co symbolizują ikony w smartfonie lub komputerze?

## Symbole a uproszczony obraz rzeczywistości

Jak powstają ikony? Poprzez uproszczenie wizerunku (obrazka). W zależności od wielkości ikonki można pokazać więcej lub mniej szczegółów. Poniżej

mamy szereg symboli kaczki:



Zabawa kształtująca abstrakcyjne myślenie: karty z rysunkami zwierząt i rzeczy - coraz bardziej abstrakcyjnych. Na jednej stronie karty (zakrytej) jest na przykład zdjęcie psa. Na drugiej (odkrytej) kolejne jego schematyczne przedstawienia – aż do najbardziej abstrakcyjnych. Zadaniem jest pogrupowanie kart z takimi rysunkami. Po odwróceniu widzimy, czy w grupach mamy takie samo zdjęcie. Takie karty można przygotować samemu posługując się wzorami ze strony http://otwartaedukacja.pl/programowanie/

Cele lekcji: 1. Zrozumienie określeń: 2. Rozpoznawanie symboli jako konkret, abstrakcja, symbol. uproszczonego obrazu rzeczy.



Temat 2. Uczymy się myśleć abstrakcyjnie.

#### DO CZEGO SŁUŻĄ SYMBOLE

#### Symbol informuje

Symbol przekazuje nam wiadomość. Jeśli ją rozumiemy – mówimy, że symbol jest zrozumiały, albo że ma znaczenie. Symbol może też na coś wskazywać. Na przykład kółko na drzwiach oznacza ubikację dla dziewczynek.

> Często spotykanymi symbolami są znaki drogowe. Zadanie: opisać na co wskazuje i jakie ma znaczenie (co znaczy, jak wpływa na oglądających go) znak drogowy z ilustracji obok.

#### Symboliczne oznaczenie stanów

Stany urządzeń mogą być oznaczane symbolami. Na przykład często są używane symbole stanu włączenia (zielone kółko) i wyłączenia (czerwona kreska). Przełącznik oznacza się kreską w kółku (dwa w jednym).

W większości urządzeń ilość stanów jest bardzo duża i opisując je stosujemy graficzne symbole tylko wtedy, gdy chcemy ułatwić użytkowanie (ikony w smartfonie).

**Pytanie:** na co wskazują ikony w smartfonie?

Zadanie: Ułóż plan wycieczki posługując się symbolami (autobus, hotel, górska wycieczka, etc....). Można wykorzystać symbole z arkusza http://otwartaedukacja.pl/programowanie/trip.pdf

#### Cele lekcji

- 1. Zrozumienie **znaczenia** symboli.
- 2. Planowanie z użyciem symboli.

#### Podsumowanie tematu

1. Symbole to abstrakcyjne (nie związane wprost z jednym konkretnym przedmiotem) przedstawienie rzeczy.

2. Symbole mogą być wykorzystywane w komputerach w postaci ikon.









Temat 3. Cyfryzacja

#### **CYFROWY - CZYLI DOKŁADNY?**

#### Jak osiągnąć precyzję?

**Precyzyjny** opis, to opis z dużą dokładnością - nie pozostawiający wątpliwości. Na przykład polecenie "wsyp trochę mąki" jest nieprecyzyjne - bo nie wiemy ile to jest "trochę". Wsyp szklankę mąki – to opis precyzyjny.

Jak coś precyzyjnie opisać? Na dwa sposoby:

1. Poprzez drobiazgowy opis najdrobniejszych szczegółów.

2. Pomijając szczegóły których precyzyjne opisanie jest trudne.

**Cyfryzacja** to wybór tego drugiego sposobu. Na przykład zdjęcia robimy z pewną **rozdzielczością** – czyli aparat nie utrwala szczegółów obrazu niedostrzegalnych dla człowieka. Są one nieistotne.

#### Cyfrowa informacja

Cyfrowy zapis obrazu lub dźwięku polega na podzieleniu go na drobniutkie fragmenciki, których może być bardzo dużo, ale nie tak dużo – by nie dało się ich policzyć. Gdy obraz zaczniemy powiększać – widać efekty uproszczenia (pominięcia nieistotnych szczegółów). Kropeczki stają się kwadratami – a kształty tracą na regularności.

Obok pokazano czerwono-niebieską kropkę i jej fragment (lewy górny róg) po powiększeniu 2 i 4 krotnym:

Posługując się cyfrowym zapisem godzimy się z tym, że część informacji staje się nieważna. Jeśli chcemy mieć obraz z większą dokładnością – to musimy podzielić go na mniejsze fragmenciki (większa rozdzielczość). Jednak zawsze późniejsze powiększanie ujawni cyfrową jego naturę.

To samo dotyczy czasu, który zostaje podzielony na fragmenty. Zadanie: Jakie znamy fragmenty czasu? (Godzina, minuta, sekunda....) W komputerze są to dużo mniejsze jednostki czasu, niż te które znamy z kalendarzy i zegarków.

Zabawa w cyfrowy czas: wykonujemy coś (spacer) wykonując ruch dokładnie co 5 sekund.

Załóżmy jednak, że mamy komputer w którym podstawową jednostką jest sekunda. Nie miałoby wówcza sensu pytanie co się dzieje między pierwszą a drugą sekundą. W międzyczasie nic się nie zmienia (wraz z każdą zmiana pojawia się kolejny moment czasu). Mówimy, że czas nie jest ciągły.

#### Zabawa w cyfryzację

Dzielenie obrazków na drobne fragmenciki nie pojawiło się wraz z komputerami. Dużo starszym pomysłem jest haft krzyżykowy. Poniżej (po lewej) projekt kwiatka i wykonany według tego projektu haft:



Komputery sprawiły, że te stare sposoby nabrały zupełnie nowego znaczenia. Ciekawy przykład ery komputerów:



Po dużym pomniejszeniu można rozpoznać jeden z najsłynniejszych arcydzieł malarstwa (więcej: http://otwartaedukacja.pl/programowanie/).

#### Ćwiczenia do samodzielnego przygotowania:

1. Układanki z kwadracików (lub kółeczek) które można powiększać i pomniejszać obraz (patrzeć z różnej odległości). Można wykorzystać tapety ze strony https://www.brik.co/blogs/pixel-art

2. Rysujemy samemu obrazki w postaci kwadracików / pikseli: http://makepixelart.com/free/#

#### <mark>Cele lekc</mark>ji

Zrozumienie, że obraz "cyfrowy" jest bardzo dokładny ale też uproszczony.

#### PRZETWARZANIE CYFROWEJ INFORMACJI

Cyfrowa natura informacji sprawia, że łatwiej ją przetwarzać automatycznie. **Instrukcja** jak pomalować kwadraciki na kartce papieru może być bardzo precyzyjna. Na przykład poniższa kartka zawiera cyfry. Malując je odpowiednimi kolorami dostajemy rysunek bohatera kreskówek:

	A	A	A	A	A	<i>.</i>	<i>.</i>			A	<i>b</i>		A	<i>.</i>	<i>.</i>	A		A	
Spec	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	6	6	6	6	6	6	6	6	1	1	1	1	1	6	6	6	6	6	6
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	6	6	6	6	6	6	6	1	1	2	2	1	1	6	6	6	6	6	6
	6	6	6	6	6	6	6	2	1	2	2	2	2	6	6	6	6	6	6
	6	6	6	6	6	6	6	2	2		1	3	1	6	6	6	6	6	6
Key:	6	6	6	6	6	6	6	2	2	2	2	3	2	6	6	6	6	6	6
Black Black Light Beige Dark Beige Blank squares are with	6	6	6	6	6	6	6	2	2	2	2	3	2	6	6	6	6	6	6
	6	6	6	6	6	6	6	2	2	2	2	2	2	6	6	6	6	6	6
	6	6	5	5	5	5	5	3	2	2	2	2	2	5	5	5	5	5	6
	6	8	8	7	7	5	5	5	3	3	3	3	5	5	5	7	7	8	8
Pomaluj	8	7	7	8	7	7	7	7	7	7	7	7	7	7	7	7	8	7	7
1 Стокры	7	7	7	7	7	7	5	5	5	5	5	5	5	7	7	7	7	7	7
2 Jasny F	7	7	7	7	7	5	4	5	5	5	5	5	4	5	7	7	7	7	7
3 Ciemny	7	7	8	7	5	4	4	4	4	4	5	5	5	5	5	7	8	7	7
4 Żółty	7	8	5	7	7	5	4	4	5	5	5	4	4	5	7	7	5	8	7
5 Ciemna	7	8	5	8	7	7	5	5	5	5	4	4	5	7	7	8	5	8	7
6 Jasho I 7 Niebies	7	8	5	8	7	7	7	5	5	5	5	5	7	7	7	8	5	8	7
8 Ciemno	8	5	5	5	7	8	7	7	5	4	5	7	7	8	7	5	5	5	8
	8	5	5	5	8	7	8	7	7	5	7	7	8	7	8	5	5	5	8
	8	5	5	5	8	7	7	8	7	7	7	8	7	7	8	5	5	5	8
	-	*		-	2	×	-	-	-	-	-	-	-	-	-	-	-	*	-



#### według numerów:

- Beż
- / Beż
- a czerwień
- niebieski
- ski
- o niebieski

źródło (więcej: https://www.coloringsquared.com/worksheet): https://www.coloringsquared.com/wp-content/uploads/2016/07/2-Superman-DC-Comic-Coloring.pdf

Dzięki uproszczeniu obrazu, możemy go łatwo opisać symbolami – tu znakami poszczególnych cyfr od 1 do 8.

Instrukcje typu "pomaluj kratki z cyframi" jest wystarczająca dla człowieka. Mamy dużą swobodę działania (szybkość, kolejność, dobór kolorów). Instrukcja dla automatu musi być bardziej precyzyjna.

Zabawmy się w robocika.

Mamy ponumerowane kratki (rysunek obok). Można je narysować kredą na podłodze. Napiszmy bardzo precyzyjnie jak je zamalować (przykryć kolorową kartką) chodząc od kratki do kratki - tak aby osiągnąć szachownicę (rysunek poniżej).

Opis taki może wyglądać mniej więcej tak:

1. Wejdź do kratki oznaczonej cyfrą "1"

2. Maluj

3. Przejdź do kratki oznaczonej cyfrą "2"

4. .....

Teoretycznie możemy skakać po kratkach – ale prościej jest założyć, że możemy przechodzić od kratki tylko do sąsiedniej kratki.

Nasze kwadraciki mogą nie być na początku puste, ale zawierać instrukcje działania.

Przekształćmy nasz program na zdania typu (zakładamy, że na początku automat przechodzi do kratki "1"):

1. Jeśli jesteś w kratce oznaczonej cyfrą "1" to maluj i przejdź do kratki "2". 2. ....

W takim programie automat czyta co jest w kratce (cyfrę) i na tej podstawie wykonuje działanie. Czyli cyfry zawierają instrukcje!

Możemy także zaznaczać gwiazdkami kwadraciki do pomalowania:

Nasza instrukcja może przyjąć formę:

1. Jeśli w kratce jest gwiazdka to maluj.

2. Przejdź do kratki sąsiedniej.

Tak mniej więcej wyglądają instrukcje w programach komputerowych. Zabawa w programowanie może mieć

formę odgrywania roli robota, albo posługiwania się prawdziwym robotem (OzoBot'y https://edu-sense.com/pl/lekcje/)

## Cele lekcji

Program jako **instrukcje** dla **automatu**. Poznajemy sposób w jaki symbole (tu: cyfry) służą do układania programów.

#### Podsumowanie tematu

 Cyfrowa dokładność polega na precyzji opisywania i zmian – ale uproszczonego obrazu. To co pomijamy staje się nieważne.
Ta precyzja sprawia, że możemy tworzyć równie precyzyjne instrukcje zmiany cyfrowej informacji.

# 1\* 2 3\* 6 5\* 4 7\* 8 9\*



# **ARKUSZ KALKULACYJNY**

Żaden automat ani komputerowy program nie działa sam z siebie. Trzeba go uruchomić. Czyli zmienić jego stan na "uruchomiony". Także później możemy modyfikować działanie zmieniając stan (ustawienia). Urządzenia i sposoby umożliwiające nam takie modyfikacje nazywamy **interfejsem**.

Interfejs to coś, co umożliwia człowiekowi zmianę stanu urządzenia. Na przykład przycisk w windzie powodujący otworzenie jej drzwi, lub ekran smartfona, którego dotknięcie powoduje wykonanie połączenia.

W naszej zabawie w malowanie kratek (z poprzedniej lekcji) możemy modyfikować działanie "robota" mażąc lub rysując gwiazdki w poszczególnych kratkach (po których porusza się robot).

Interfejs w postaci kratek jest zastosowany w jednym z najbardziej popularnych programów komputerowych: arkuszu kalkulacyjnym. Ideą tego programu jest wyliczanie wartości wpisanych w poszczególnych kratkach na podstawie zawartości innych kratek wskazanych przez użytkownika. Oznaczmy dla odmiany kratki kolorami, a cyfry niech oznaczają wyliczoną wartość. Prosty przykład pokazujący definiowanie powiązań przy pomocy kolorów zawiera program: http://otwartaedukacja.pl/programowanie/kss/



Komórki w drugim rzędzie mają kolorowe rogi, wskazujące na inne komórki (o takich samych kolorach co te rogi). Wartość wyświetlana w komórce to suma wartości zapisanych w rogach, a ta zmienia się wraz ze zmianą koloru. Na przykład wartość wyliczona w drugiej z lewej (zielonej) komórce to suma komórki niebieskiej i fioletowej. Zmiany powiązań dokonujemy w dwóch krokach:

- kliknij w komórkę którą chcesz wskazać (zmieni ona kolor na niebieski)

- kliknij w róg, który ma na wybraną komórkę wskazywać.

Powszechnie używane arkusze kalkulacyjne wymagają wpisywania liczb i **wyrażeń** (takich jak 2+4) ręcznie. W tym celu rzędy komórek oznacza się liczbami (1,2,3,..) a kolumny literami (A,B,C,...). Jeśli w powyższym przykładzie mamy rzędy 1 i 2, to do zielonej komórki powinniśmy wpisać A2+A4 . Arkusze kalkulacyjne mogą wykonywać **obliczenia** bardziej skomplikowane, ale podstawowa idea jest taka sama: komórki są powiązane ze sobą i na tej podstawie wylicza się ich wartość.

Takim arkuszem może być program LibreOffice Calc, albo na przykład dokument Google: https://docs.google.com/spreadsheets.

fх	= <mark>a1</mark> +b1		
	А	В	С
1	3 ×	1∎ 2∎	
2	= <mark>a1</mark> +b1		
3			
4			
5			

#### Cele lekcji

1. Rozumienie stanu jako **wartość** (na przykład wyrażenia arytmetycznego). Ustalenie tej wartości to **obliczenia.** 

 Ważne pojęcia: użytkownik, interfejs, wyrażenie, obliczenia, arkusz kalkulacyjny.



# **KALKULATOR**

W czasie poprzedniej lekcji poznaliśmy bardzo praktyczny sposób automatycznych obliczeń, spotykany w wielu dziedzinach: arkusz kalkulacyjny. Istnieje jednak urządzenie cyfrowe jeszcze prostsze – kalkulator. Prosty przykład kalkulatora znajdziesz na stronie: http://otwartaedukacja.pl/programowanie/calculator/ Wpisywane cyfry tworzą liczbę (pole niebieskie) a znak + lub – powoduje utworzenie wyrażenia (pole czarne) – czyli działanie dodawania lub odejmowania. Wynik tego wyrażenia jest od razu zapisywany w polu czerwonym.



Takie **wyrażenia** mogą być zapisywane w komórkach arkusza kalkulacyjnego – po znaku '=' (na przykład =4+5). Zamiast liczb możemy wskazywać inne pola. Wpisujemy = a później klikamy w pole z którego należy pobrać wartość.

### Zadanie:

Wpisz w arkuszu kalkulacyjnym wyrażenie takie jak z kalkulatora i porównaj wyniki.

Takie same wyrażenia stosuje się w programowaniu. Są podstawową częścią tak zwanych "**języków programowania**". Jeśli chcemy przekazać cokolwiek do wykonania komputerowi – nie mówimy do niego po polsku lub angielsku, ale w języku wymyślonym do takiej komunikacji (takim sztucznym językiem jest esperanto – ale on ma ułatwić komunikację między ludźmi, a nie z komputerem).

Jeden z takich języków nosi nazwę Python (po polsku Pyton). Zadanie: spróbuj wpisać wyrażenie do czarnego okienka na stronie

https://www.python.org/shell/ i naciśnij Enter. Co się stało?

## Cele lekcji

1. Poznajemy wyrażenie jako zapis (ślad) działania kalkulatora.

2. Ważne pojęcia kalkulator, wyrażenie, język programowania.

#### Podsumowanie tematu

1. Poznajemy dwa podstawowe programy komputerowe: kalkulator i arkusz kalkulacyjny.

2. Wyrażenia arytmetyczne występujące w powyższych programach są podstawowym elementem wszystkich języków programowania.

Działania sekwencyjne i żółw

# GRAFIKA ŻÓŁWIA

W jakiej kolejności wykonywane są operacje przez komputer (maszynę)? Po kolei – jedna po drugiej. Inaczej mówiąc **"sekwencyjnie**". Do zilustrowania takiego działania możemy wykorzystać "grafikę żółwia". Wyobraźmy sobie żółwia z przywiązanym mazakiem (pen). Chodząc po planszy może on rysować różne wzory.

W lekcji 4 planowaliśmy wycieczkę. Podobnym zadaniem może być zaplanowanie wycieczki dla żółwia po kwadracie

#### http://otwartaedukacja.pl/programowanie/turtle/

Napiszmy dla takiego żółwia instrukcję narysowania prostokąta 100 x 100:

- opuść pisak
- idź 100 jednostek
- skręć w prawo o 90stopni
- idź 100 jednostek
- skręć w prawo o 90stopni
- idź 100 jednostek
- skręć w prawo o 90stopni
- idź 100 jednostek



Zamiast pisania instrukcji składa się ją z klocków dostępnych w menu po lewej stronie. Po ustawieniu całości klikamy w miejsce oznaczone "start" i żółw zaczyna rysować.

W zakładce oznaczanej piórem są "klocki" z kolorami. Kolory są oznaczone kolejnymi liczbami (0, 1, 2, ....).

Zadanie: narysować kwadrat w którym każdy bok jest innego koloru.

#### Cele lekcji:

Poznajemy ideę działania **sekwencyjnego** oraz "grafikę żółwia".

#### ZMIENNA

W arkuszu kalkulacyjnym wyrażenia (reguły) i wynik zapisywane są w tej samej komórce. Każda komórka arkusza kalkulacyjnego ma swoją nazwę utworzoną z numeru kolumny i wiersza (lewy górny róg to A1). Inne słowo – często używane zamiast "nazwa" to "**identyfikator**". Identyfikator to nazwa która identyfikuje jeden z wielu elementów. Komórkę pamięci opatrzona identyfikatorem nazywamy "**zmienną**" (co ma podkreślić fakt, że jej zawartość się zmienia). To jest pojęcie często używane w programach – nie tylko w arkuszach kalkulacyjnych.

Zmienna to informacja, którą komputer/urządzenie zapisuje "na potem" - np. arkusz kalkulacyjny zapisuje wartość komórek, żeby móc je wyświetlić użytkownikowi lub użyć w obliczeniach, winda zapisuje numer piętra, na które ma się przemieścić.

Pomocnym w zrozumieniu pojęcia zmiennej jest popularna gra "Memo" w odgadywanie co kryje się w jednakowych polach kratki (na przykład: http://prek-8.com/games/memoryGames/memoryGames1.php). Gdy oznaczymy literkami – odgadnięcie staje się łatwiejsze, nawet gdy w kolejnym kroku zmienimy losowo układ pól.

W przypadku grafiki żółwia możemy także tworzyć zmienne. Zostały one nazwane w tym programie "pudełkami" ("boxes") i są oznaczane jasnobrązową barwą. Utworzenie pudełka (zmiennej) wykonujemy przez przeciągnięcie i puszczenie klocka "przechowaj w". Następnie klikamy w nazwę "zmiennej" (domyślnie "pudło") i wpisujemy dowolny wyraz. Na przykład "barwa". Tak utworzoną zmienną (pojawi się w zakładce pudełek) możemy użyć tak jak liczb. Czyli wstawiając ją do zmiennej przechowującej kolor dokonujemy zmiany koloru na taki, jaki mamy zapamiętany w zmiennej "barwa". Objaśnia to poniższy przykład (na następnej stronie).

#### Zastosowanie zmiennej / pudełka



Aktualny kolor także jest pamiętany w zmiennej, ale nie jest ona wyświetlana. Wykonując operację "ustaw kolor" zmieniamy tą zmienną.

#### Cele lekcji

- 1. Poznajemy sposób operowania na komórkach pamięci / **zmiennych.**
- 2. Ważne pojęcia **zmienna** (na przykład komórka arkusza kalkulacyjnego).

#### Podsumowanie tematu

1. Poznajemy grafikę żółwia, która pomaga nam w zrozumieniu idei działania sekwencyjnego.

2. Zmienne to komórki pamięci komputera w których zapamiętuje się wartości. Każda zmienna posiada swój identyfikator.

# LOGIKA I ŚWIAT WIRTUALNY

Wyobraźmy sobie, że jesteśmy żółwiem z naszego programu do rysowania, który poznaliśmy w poprzednich lekcjach. Taki żółw dostrzega jedynie to co mu ustawimy na planszy i co sam narysuje. Jego świat ogranicza się do planszy którą może wypełniać kreskami oraz klocków tworzących plan działania. Tego rodzaju wymyślony świat nazywa się **światem wirtualnym** (nie istniejącym realnie tak jak świat w którym my żyjemy).



Przykładem takiego wirtualnego świata może być Kraina Czarów w której znalazła się bajkowa Alicja.

Zadanie: jakie jeszcze znamy bajkowe "światy wirtualne"?

Czym różni się kraina w której zamieszkuje rysujący kreski żółw od "krainy czarów"? W krainie czarów możliwe jest prawie wszystko. Tymczasem żółw ma bardzo ograniczone możliwości działania. Takie ograniczenia dotyczą każdego komputerowego "świata wirtualnego".

#### Programowanie komputerów to budowanie wirtualnych światów!

Tak jak w naszym świecie rządzą prawa przyrody, w komputerowych światach wirtualnych rządzą prawa logiki. Znaczy to, że z prawdziwej informacji zawsze wynika prawda. Dochodzimy do tej prawdy drogą wnioskowania.

Na przykład jeśli mama była na zakupach i jest dzień dziecka, to możemy się spodziewać prezentu. Jest to przykład "wyciągania wniosków", czyli **wnioskowania**. Warunki z których wyciągamy wnioski nazywa się "**przesłankami**" (tu: zakupy i dzień dziecka).

Ćwiczenie: spróbujmy samodzielnie wymyślić podobne przykłady wnioskowania.

Programowanie wiąże się z wnioskowaniem. Podobnie jak rozwiązywanie logicznych łamigłówek. Dlatego takie łamigłówki to doskonałe ćwiczenie dla przyszłych programistów. Przykłady gier logicznych: http://www.learninggamesforkids.com/logic\_games/sliding\_block\_puzzle.html

Logika i świat wirtualny

# WNIOSKOWANIE W PROGRAMOWANIU

Przypomnijmy sobie program rysowania szachownicy z lekcji 5. Jak bardzo uprościł się nasz opis, gdy wprowadziliśmy gwiazdki do oznaczenia kwadratów do zamalowania!

Program sprowadza się do "instrukcji warunkowej":

Jeśli w kratce jest gwiazdka to maluj i przejdź do następnej kratki a jeśli gwiazdki nie ma to tylko przejdź do następnej kratki.



Każdą instrukcję działania (każdy program) teoretycznie można zapisać w taki sposób. Spróbuj opisać włączanie laptopa posługując się takimi warunkowymi instrukcjami:

- jeśli klapa jest zamknięta, to ...

- ....

W praktyce większość instrukcji nie musi być warunkowa, bo wykonujemy je sekwencyjnie (po kolei). Nie musimy pisać: jeśli otworzyłeś klapę laptopa, to naciśnij przycisk, skoro poprzednia instrukcja nakazywała otwarcie klapy.

Związek logiki i programowania powinien być już łatwy do zrozumienia! Rozwiązując problem wyciągamy wnioski z przesłanek. Zapisując to rozwiązanie w postaci programu, używamy instrukcji warunkowych lub sekwencji instrukcji.

Komputer wykonując ten program działa automatycznie: jeśli jest w pewnym stanie (określonym przez wartość zmiennych) to przechodzi do stanu następnego (następnej instrukcji).

Przeanalizujmy to na przykładzie trójkąta równobocznego. Jest to figura o trzech równych bokach. Możemy zauważyć, że kąty między bokami w trójkącie równobocznym są jednakowe. A ponieważ w geometrii stosuje się miarę według której suma kątów w każdej wypukłej figurze wynosi 360 stopni, kąty w trójkącie równobocznym muszą mieć po 120 stopni. Znając zasady grafiki żółwia możemy już napisać dla niego program.



**Cele lekcji**: Pokazanie na przykładach jak stosujemy wnioskowanie w algorytmach i programowaniu

#### **AKCJE I POWTÓRZENIA**

Wróćmy ponownie do naszego programu z grafiką żółwia. Program ten wyświetla zapamietany (w zmiennych) układ klocków i położenie żółwia Po uruchomieniu programu system (program) zmienia stan: rysuje i przesuwa żółwia.

Dodatkowo są używane zmienne (niewidoczne) w których przechowuje się aktualny kolor rysowania i wskazuje na aktualnie wykonywane polecenie rysowania (bloczek programu). Klikając w ikonkę ślimaka, możemy obserwować wykonywanie programu "krok po kroku" (już wykonane instrukcje są wyświetlone jaśniejszym kolorem).

Budując coraz bardziej złożone rysunki, otrzymujemy gmatwaninę klocków trudną do zrozumienia. Można tego uniknąć stosując powtórzenia i akcje. Najpierw spróbujmy wykorzystać akcje (w językach programowania nazywane funkcjami). Są to fragmenty programu oznaczone nazwą. W programie z grafiką żółwia akcje znajdują się w zakładce otwieranej ikonką z dwoma zawiniętymi strzałkami. Zdefiniujmy dla przykładu akcję "kwadrat" - przesuwając do niej wszystkie instrukcje.



Po zdefiniowaniu (zmianie nazwy) w zakładce akcji pojawia się klocek "kwadrat", który możemy użyć w miejsce całej sekwencji działań (klocek "start").

Wszędzie gdzie pojawi się klocek "kwadrat" zastępuje on całą sekwencję działań. Jeśli ta sekwencja (rysowanie kwadratu) ma się pojawić tylko raz – to zastosowanie akcji nie jest wielkim uproszczeniem. Ale nawet w takim przypadku program dzięki użyciu akcji może być bardziej zrozumiały.

Drugim ważnym mechanizmem są powtórzenia. Gdy chcemy narysować 4 kwadraty, to zamiast układać jeden pod drugim cztery razy "kwadrat" możemy użyć klocka "powtarzaj":



## **ALGORYTM A PROGRAM**

**Algorytmy** to dokładny opis sposobu postępowania prowadzącego do określonego wyniku.

Na przykład algorytm otwierania drzwi: aby otworzyć drzwi, włóż klucz do zamka, przekręć w prawo do oporu a następnie naciśnij klamkę.

Czym różni się algorytm od programu? Programy które tworzy programista są zawsze przeznaczony do wykonania przez automat (komputer). Musi więc być szczegółowy i wyrażony w instrukcjach dla automatu (lub w sposób możliwy do przetłumaczenia na takie instrukcje).

Algorytm może być opisany mniej szczegółowo. Na przykład wystarczającym opisem algorytmu rysowania kwadratu jest: narysuj figurę o czterech jednakowych bokach prostopadłych do siebie. Pisząc program dla rysującego żółwia musimy zdawać sobie sprawę z tego, że on "nie rozumie" słów "figura" i "prostopadły". Zadaniem programisty jest przełożenie opisu algorytmu instrukcje zapisywane w języku programowania. Te języki nie są dużo bardziej skomplikowane niż poznany przez nas "język klocków" w którym układamy trasę podróży żółwia.

Większość algorytmów jest równie prosta, jak powyższy opis otwierania drzwi. Jednak algorytmy bywają też bardzo trudne. Wiele ludzi utożsamia układanie algorytmów z programowaniem – dlatego programowanie uważa się za coś trudnego. Tak jednak nie jest – skoro każdy potrafi napisać program dla żółwia. Poniżej mamy obrazki czterech takich programów w postaci akcji o

identyfikatorach p1,p2,p3,p4).





Aby je uruchomić, trzeba odpowiednio spreparować klocek "start" na jeden z poniższych dwóch sposobów (w miejsce p3 trzeba wstawić odpowiednią akcję)



Spróbuj dopasować do którego z algorytmów pasuje który program?

Nazwa	Algorytm Program (p1,p2,p3,p4)?
Kółko	Narysuj łuk o zadanym promieniu i kącie pełnym (360 stopni)
Kwadra	Narysuj figurę o czterech jednakowych bokach prostopadłych do siebie (kąt 90 stopni)
Trójkąt	Narysuj figurę o trzech jednakowych bokach
Krzyż	Narysuj figurę składającą się z dwóch kresek przecinających się w połowie pod kontem prostym (90 stopni).

Jeśli masz problem z dopasowaniem – możesz uruchomić program

(http://otwartaedukacja.pl/programowanie/turtle/) i zbudować program, lub odczytać go z "**biblioteki**" (ikona globu, arkusz "figury").

Takie biblioteki z gotowymi akcjami są powszechnie dostępne w internecie – dla różnych języków programowania i przeróżnych algorytmów. Umiejętność posługiwania się nimi jest jedną z najważniejszych dla programistów.