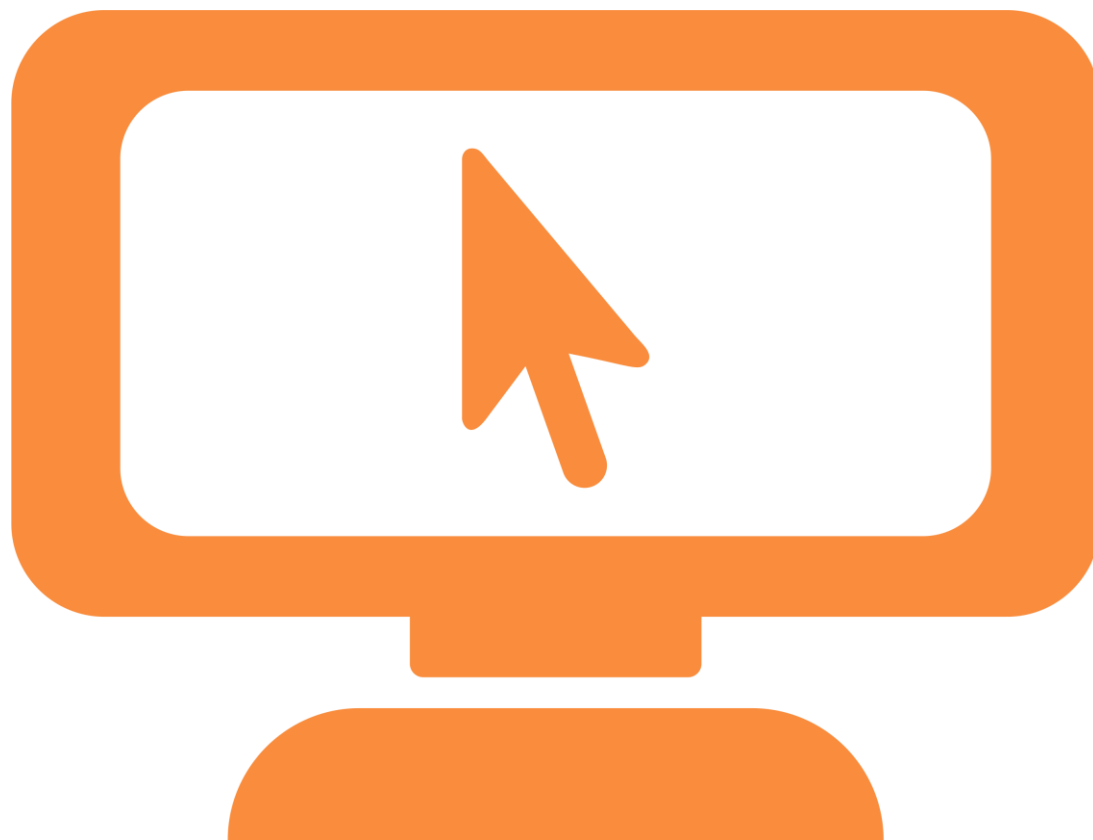


| Prywatne Gimnazjum Językowe PARNAS |



**Skrypt metodyczny do
programu nauczania informatyki
poszerzonej o elementy robotyki**

SPIS TREŚCI

1. Wstęp, koncepcja i charakterystyka programu	2
2. Warunki realizacji	2
2.1. Wyposażenie pracowni	2
2.2. Oprogramowanie	3
3. Wykaz zagadnień, formy i metody pracy	3
3.1. Systemem operacyjny.....	3
3.2. Budowa komputera oraz zasadę działania urządzeń zewnętrznych.....	3
3.3. Sieci komputerowe	4
3.4. Edytor grafiki wektorowej	4
3.5. Edytor grafiki rastrowej.....	4
3.6. Edytor tekstu	5
3.7. Arkusz kalkulacyjny – wstęp.....	5
3.8. Arkusz kalkulacyjny – zadania zaawansowane	6
3.9. Bazy danych.....	7
3.10. Strona internetowa – HTML + CSS	8
3.11. Algorytmy – informacje wstępne, schematy blokowe	9
3.12. Programowanie robota LEGO w środowisku graficznym	10
3.13. Język C++ - podstawy	11
3.14. Język C++ - programowanie zaawansowane	12
3.15. C++ - wybrane algorytmy klasyczne	14
4. Przewidywane osiągnięcia uczniów	14
5. Sposoby oceniania.....	16

1. Wstęp, koncepcja i charakterystyka programu

Skrypt metodyczny do programu nauczania „Komputer narzędziem pracy parnasisty i ty możesz zostać programistą” dla uczniów klas matematyczno-informatycznych Prywatnego Gimnazjum Językowego Parnas ma na celu pokazanie, w jaki sposób mogą być realizowane cele zakładane w programie nauczania. Opracowany program miał dwa zasadnicze cele. Pierwszy to pokazanie młodemu człowiekowi możliwości, jakie daje postępująca informatyzacja. Każdy uczeń powinien umieć wykorzystać komputer do osiągnięcia zamierzonych celów w przyszłości i mieć świadomość, że cel ten można osiągnąć szybciej, sprawniej i taniej przy pomocy komputera. Kolejny cel to zapoznanie uczniów z szeroko rozumiana algorytmiką, programowaniem robotów LEGO, poznanie języka C++, omówienie wybranych algorytmów klasycznych. Dodatkowo uczniowie poznają wybrane programy do tworzenia i edycji grafiki komputerowej oraz projektują stronę Internetową.

2. Warunki realizacji

Program może być realizowany w szkolnej pracowni komputerowej wyposażonej w komputery oraz zestawy LEGO. Zgodnie z § 10 rozporządzenia MEN z dnia 7 lutego 2012 r. w sprawie ramowych planów nauczania w szkołach publicznych (Dz. U. z 2012 r., poz. 204), na jednego ucznia musi przypadać jeden komputer. Na każdym komputerze uczniowie posiadają swojego użytkownika, co jest ważne przy pracach długo terminowych, gdzie konieczny jest zapis postępujących prac. Wszystkie komputery powinny posiadać dostęp do sieci Internet oraz posiadać niezbędne oprogramowanie.

2.1. Wyposażenie pracowni:

- komputery MS Windows – po jednym na każdego ucznia
- sieć komputerowa z dostępem do Internetu i zasobów lokalnych
- rzutnik
- LEGO MINDSTORMS Education EV3 – zestawy edukacyjne LEGO

2.2. Oprogramowanie:

- System operacyjny Windows
- Microsoft Office – MS Word, MS Excel, MS Access, MS Power Point
- Oprogramowanie graficzne do tworzenia grafiki wektorowej - Corel Draw
- Oprogramowanie graficzne do edycji grafiki rastrowej – Adobe Photoshop
- Oprogramowanie LEGO Mindstorms – graficzne programowanie robotów
- Kompilator języka C++
- Program do tworzenia animacji komputerowych Adobe Flash.

3. Wykaz zagadnień, formy i metody pracy

3.1. Systemem operacyjny

W tej części należy zwrócić uwagę na podział systemów operacyjnych typu Windows i Unix. Należy zapoznać uczniów ze strukturą folderów na komputerze. Przećwiczyć kopiowanie, przenoszenie, tworzenie skrótów. Wprowadzając pojęcie pliku trzeba pokazać, że poza nazwą istnieje rozszerzenie. Część zajęć trzeba poświęcić na wyszukiwanie plików danego typu w systemie operacyjnym. Można zwrócić uwagę na konta użytkowników, bezpieczeństwo danych.

3.2. Budowa komputera oraz zasady działania urządzeń zewnętrznych

Lekcja teoretyczna, w czasie której uczniowie zapoznają się z budową komputera. Bardzo dobrym pomysłem jest przyniesienie na lekcję wybranych podzespołów i pokazanie ich uczniom. Zapoznujemy uczniów z urządzeniami zewnętrznymi, budową płyty głównej, wymieniamy pamięci zewnętrzne i wewnętrzne. Wprowadzamy jednostki pamięci komputera oraz ich przeliczanie.

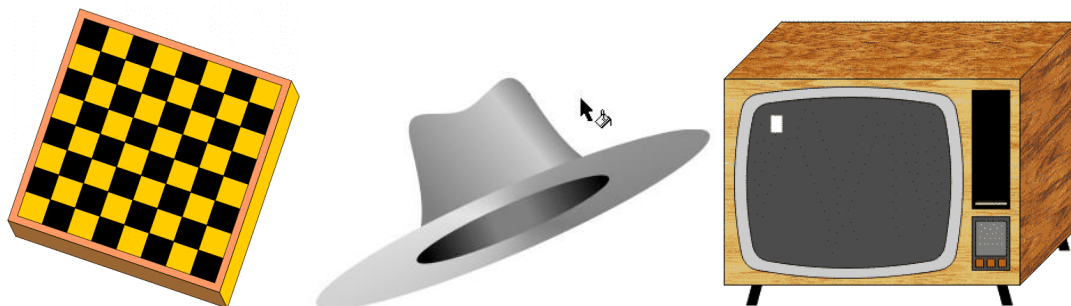
3.3. Sieci komputerowe

Wprowadzamy pojęcie sieci jako połączenie urządzeń. Zapoznajemy uczniów z rodzajami połączeń – kablowe i bezprzewodowe. Należy zwrócić uwagę uczniów na adresowanie w sieci, pokazanie wybranych logów serwera uświadamia uczniów, że w sieci nie jest się anonimowym. Omawiamy, jak wygląda witryna z szyfrowanym połączeniem, zwracamy uwagę na bezpieczeństwo – zapamiętane hasła, podmiana witryny, linki w wiadomościach e-mail.

3.4. Edytor grafiki wektorowej

Uczniowie pracują na najbardziej popularnym edytorze grafiki wektorowej jakim jest Corel Draw. Warto zacząć zajęcia od pokazania prostych kompozycji z figur np. bryła sześcianu. Uczniowie uczą się, jak ścinać, obracać, wypełniać figury. Kolejne godziny można poświęcić na naukę krzywych Beziera. Jak ustawiać węzły i manipulować punktami kontrolnymi, aby otrzymać wybrany kształt. Można korzystać z pomysłów zaczerpniętych z sieci. Warto pokazać jak można podzielić obszar wydruku na części, efekty dla tekstu i figur. Edukację grafiki wektorowej należy zakończyć projektem np. samodzielnie zaprojektowana ulotka reklamowa.

Kilka przykładów prostych projektów graficznych do realizacji na lekcji:



3.5. Edytor grafiki rastrowej

Uczniowie pracują w programie Adobe Photoshop. Pierwsze zajęcia trzeba poświęcić na zmianę rozmiaru zdjęcia, prostą korektę barw, nasycenia. Kolejny temat to korekta zdjęcia przy użyciu narzędzia stempel. Gimnazjaliści pracują na warstwach, stosują filtry, zaznaczenia, wtapianie. Ostatnie zajęcia to projekt graficzny z wykorzystaniem poznanych narzędzi – tło, warstwy, filtry, zaznaczanie.

3.6. Edytor tekstu

Zapoznajemy uczniów z zasadami pisania w edytorze tekstu – nie używamy spacji i klawisza enter do przesuwania tekstu. Należy wprowadzić pojęcie akapitu i jego formatowanie – okno akapit – wcięcia, wcięcia specjalne, odstępy, wyrównania. Jedną lekcję poświęcamy tabulatorom i tabelom: do czego służy tabulator, rodzaje wyrównania (lewy, prawy, środkowy), znaki wiodące. Pracujemy na dokumencie wielostronicowym – wstawiamy nagłówek, stopkę, numerację, przypisy, podział strony oraz sekcje. Warto przygotować dokument wielostronicowy z różnym formatowaniem i polecić uczniom formatowanie zgodnie ze wzorem. Przykładowe ćwiczenie dotyczące formatowania tekstu:

The diagram illustrates text formatting in a word processor. It shows a letter and a list of references with various formatting annotations:

- Address:** Konrad Michalewski, ul. Sezamowa 5/1, 54-012 Wrocław. Annotations: "odstęp po akapicie: 18 pt" (margin after paragraph), "odstęp przed: 30 pt, po: 6 pt" (margin before/after), "wcięcie pierwszego wiersza: 1,2 cm" (first line indent).
- Text Alignment:** "Wrocław, 10 marzec 2004 r." is right-aligned. Annotation: "tekst wyrównany do prawej tabulatorem prawym – 16 cm".
- Redactor Address:** Redaktor Stefan Maczek, Gazeta Niedzienna, ul. Zabytkowa 13, 45-210 Wrocław. Annotation: "wyrównanie tekstu tabulatorem lewym – 10 cm" (left-align text with tabulator).
- Salutation:** Szanowny Panie Redaktorze,.
- Main Text:** Byliśmy sobie z klasą w muzeum Panoramy Raclawickiej we Wrocławiu. Chcę o niej napisać. Różne dane zaczerpnąłem z książek kupionych w muzeum :
 - Panorama i panoramy, Muzeum Narodowe we Wrocławiu, 1995 r.
 - Panorama Raclawice, K. M. Cwiek, Wydawnictwo ZET, Wrocław 1999 r.
- Text Indentation:** "Panorama Raclawicka jest najsłynniejszym polskim płótnem batalistycznym – przedstawia zwycięską bitwę insurekcji Kościuszkowskiej pod Raclawicami z 4 kwietnia 1794 r. Namalowano ją na Powszechną Krajową Wystawę we Lwowie w 1894 r., dla uczczenia setnej rocznicy powstania kościuszkowskiego. Malarze to byli: Jan Styka,". Annotations: "wcięcie: 2 cm", "wysunięcie: 0,5 cm", "tabulator lewy: 2 cm".
- Text Spacing:** "uczczenia setnej rocznicy powstania kościuszkowskiego." has a "twarda spacja" (hard space) annotation.
- List Indentation:** The list items have "interlinia: 1,2", "odstęp po akapicie: 6 pt", and "tekst wyjustowany" (justified text) annotations.

3.7. Arkusz kalkulacyjny – wstęp

Rozpoczęcie zajęć w tej części należy rozpocząć od obliczeń przy pomocy formuły oraz stosowanie funkcji. Zaznajamiamy uczniów z rodzajami odwołań – względne, bezwzględne i mieszane. Gimnazjaliści tworzą różne wykresy zgodnie z poleceniami. Uczniowie wykonują wiele ćwiczeń stosując formuły, funkcje i wstawiając wykresy. Ćwiczenia dostępne są w Internecie pod adresem www.parnas.teb.pl/excel.

Przykładowe ćwiczenie z arkusza kalkulacyjnego:

J14							
A	B	C	D	E	F	G	H
2	ROZLICZENIE ZAKUPU ZDJĘĆ KLASOWYCH						
3	ROZLICZENIE ZAKUPU ZDJĘĆ KLASOWYCH						
4	ROZLICZENIE ZAKUPU ZDJĘĆ KLASOWYCH						
5	lp	nazwisko i imię	zdjęcia grupowe	zdjęcia legitymacyjne	zdjęcia portretowe	zdjęcia koleżeńskie	do zapłaty
6	1	uczeń 1	1	0	3	1	
7	2	uczeń 2	3	1	4	2	
8	3	uczeń 3	1	3	2	3	
9	4	uczeń 4	0	4	1	0	
10	5	uczeń 5	1	2	0	1	
11	6	uczeń 6	2	3	0	3	
12	7	uczeń 7	4	4	0	4	
13	8	uczeń 8	5	5	0	1	
14	do zapłaty						
15							
16	cennik						
17	zdjęcia grupowe		5,23 zł				
18	zdjęcia legitymacyjne		7,48 zł				
19	zdjęcia portretowe		6,99 zł				
20	zdjęcia koleżeńskie		8,23 zł				
21							
22							
23							
24	POLECENIA						
25	Na podstawie posiadanych informacji sporządź arkusz kalkulacyjny pozwalający Ci rozliczyć zakupione przez uczniów zdjęcia.						
26	Wpisz właściwe formuły w odpowiednie komórki. Pamiętaj jednak, że powinny być one tak zapisane, aby w						
27	każdym momencie możliwa była zmiana ceny zdjęcia, a także ilości którą zamówił/kupił uczeń.						
28	Dokonaj formatowania odpowiednich komórek na Zapis walutowy .						
29	Podpowiedź: Zauważ, że część informacji znajduje się poza zasadniczą częścią tabeli (cennik), trzeba						
30	zastosować zapis dolarowy.						
31							

3.8. Arkusz kalkulacyjny – zadania zaawansowane

Część zajęć poświęcona rozwiązywaniu zadań z wykorzystaniem wcześniejszych umiejętności. Gimnazjaliści muszą wykazać się umiejętnością analizy treści i dopasowaniem odpowiednich narzędzi. Stosują funkcje jeżeli, licznik, jeżeli, funkcje tekstowe oraz daty i czasu. Wiele pomysłów można zaczerpnąć z przykładowych zadań maturalnych dostępnych na stronie centralnej komisji egzaminacyjnej

Przykładowe zadanie:

Pani Maturalna rozważa kupno samochodu. Jednym z elementów, które będzie brała pod uwagę jest typ silnika: silnik benzynowy albo silnik diesla. Samochód wyposażony w silnik benzynowy spala przeciętnie 6,9 l na 100 km, zaś z silnikiem diesla spala 4,9 l oleju napędowego na 100 km.

Aktualna cena benzyny wynosi 4,82 zł za litr, a cena oleju napędowego jest równa 4,45 zł za litr. Co 30 000 km przeprowadza się obowiązkowy przegląd techniczny samochodu z silnikiem benzynowym. Jego koszt to 650 zł. Przegląd samochodu z silnikiem diesla wykonuje się co 20 000 km. Jego koszt to 1 320 zł.

a) Oblicz koszty eksploatacji samochodu z silnikiem benzynowym i z silnikiem diesla dla przebiegu między 100 km a 450 000 km, podając wartości co 100 km przy założeniu, że koszty przeglądu i paliwa nie ulegają zmianie. Oblicz także, jakie są różnice w kosztach użytkowania tych samochodów w zdefiniowanym zakresie kilometrów.

b) W jakich przedziałach przejechanych kilometrów, różnica w kosztach eksploatacji samochodu zawiera się między 10 000 zł a 15 000 zł. c) Na wykresie punktowym zaprezentuj różnice kosztów użytkowania samochodu.

3.9. Bazy danych

Pierwsze 2 godziny zajęć należy poświęcić pracy na prostej tabeli. Zgodnie z podstawą programową uczniowie sortują i filtrują rekordy w bazie. Należy wprowadzić ćwiczenia z tabelą przestawną w MS Excel. Kolejne godziny to praca na relacyjnych bazach danych. Uczniowie na podstawie plików tworzą tabele, wyznaczają klucz podstawowy, tworzą relacje, pracują na formularzach. Najważniejszą częścią są zapytania do bazy, tworzenie kwerend.

Przykładowe zadanie z relacyjnych baz danych:

Pliki o nazwach uczniowie.txt i wyniki.txt zawierają informacje o uczniach biorących udział w olimpiadzie informatycznej i wynikach I etapu zawodów: plik uczniowie.txt – zawiera 300 wierszy z informacjami o uczniach. W każdym wierszu znajdują się następujące dane: identyfikator ucznia, imię ucznia, nazwisko ucznia, nazwa szkoły, miejscowość, klasa. W pliku dane te są rozdzielone średnikami. Przykład: 12;Jan; Kowalski;IV Liceum Ogólnokształcące;Toruń;2 plik wyniki.txt – zawiera 300 wierszy z informacjami o zdobytych punktach przez każdego ucznia z poszczególnych zadań. Za każde zadanie uczeń może otrzymać maksymalnie 100 punktów.

W każdym wierszu znajdują się następujące dane: identyfikator ucznia, liczba punktów za zad1, liczba punktów za zad2, liczba punktów za zad3, liczba punktów za zad4, liczba punktów za zad5. W pliku dane te są rozdzielone średnikami. Uczeń, który nie wysłał zadania x ($x=1..5$), za zadanie x otrzymuje 0 punktów. Przykład: 12;100;23;99;40;50 – punkty za każde zadanie ucznia o identyfikatorze 12.

Utwórz bazę danych Olimpiada informatyczna z dwoma tabelami uczniowie i wyniki. Dane importuj z plików uczniowie.txt i wyniki.txt. Utwórz odpowiednie związki (relacje) między tabelami bazy, dbając o wymuszanie więzów integralności.

Wykorzystując dane zawarte w tabelach i dostępne narzędzia baz danych wykonaj następujące polecenia

a) Podaj nazwiska i imiona uczniów, którzy łącznie zdobyli więcej niż 400 punktów.

b) Utwórz kwerendę zawierającą dane o uczniach i wynikach według wzoru: identyfikator ucznia, imię ucznia, nazwisko ucznia, nazwa szkoły, miejscowość, klasa, zad1, zad2, zad3, zad4, zad5, suma punktów. Na podstawie kwerendy utwórz raport zawierający dane o uczniach i wynikach. Zadbaj o czytelność danych w raporcie.

c) Podaj, ilu uczniów jest z Warszawy?

d) Podaj nazwiska i imiona uczniów, którzy przechodzą do II etapu, jeśli warunkiem przejścia jest zdobycie co najmniej 250 punktów.

e) Podaj ilu uczniów z każdej miejscowości przeszło do II etapu. Uwzględnij tylko te miejscowości, które mają co najmniej 1 ucznia.

3.10. Strona internetowa – HTML + CSS

Wprowadzamy uczniom pojęcie znacznika, pokazujemy kod strony HTML. Podajemy i ćwiczymy znaczniki do formatowania tekstu, tworzymy listy, hiperłącza, formularze, tabele. Wstawiamy grafikę – warto wykorzystać wcześniejsze umiejętności z zajęć z Corel Draw lub Adobe Photoshop. Po ćwiczeniach z HTML należy uczniów zapoznać z CSS – opisać kilka znaczników zmieniając krój czcionki, kolory, marginesy, obramowania. Warto podać linki, które kierują do opisu znaczników przy pomocy CSS. Ostatnim etapem jest projekt własnej strony internetowej. Wprowadzamy znacznik <DIV>, zapoznajemy uczniów z możliwością budowy szkieletu strony www w oparciu o ten znacznik. Następnie gimnazjaliści projektują nagłówek, wstawiają linki oraz tworzą witrynę. Całość umieszczamy na serwerze stron www, warto wrócić do informacji o systemach operacyjnych zwracając uwagę na wielkość liter w systemie Linux.

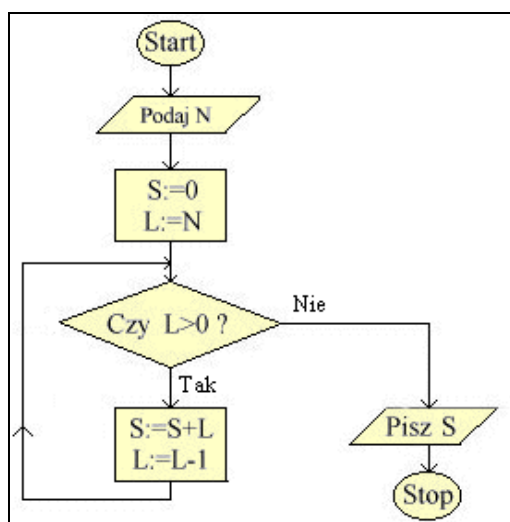
Przykładowy projekt strony www:



3.11. Algorytmy – informacje wstępne, schematy blokowe

Wprowadzamy pojęcie algorytmu, rysujemy schematy blokowe rozwiązując wybrane zadania z wykorzystaniem instrukcji warunkowych i pętli.

Przykładowy schemat blokowy i zadania:

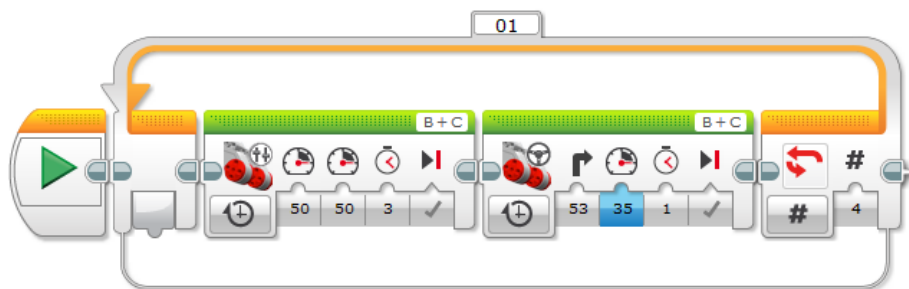


Pan Kowalski zbudował ścianę z cegieł. Każda cegła ważyła 2 kilogramy. U podstawy ściany znajdowało się 15 cegieł. Każda warstwa wyższa posiadała o jedną cegłę mniej. Opracuj algorytm za pomocą schematu blokowego, który oblicza masę ściany. Wykorzystaj instrukcję pętli.

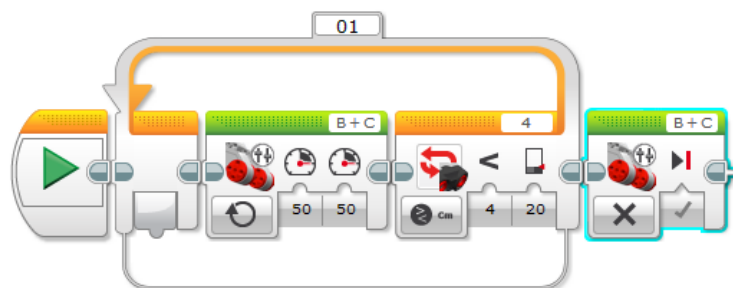
3.12. Programowanie robota LEGO w środowisku graficznym

Zaznajamiamy uczniów z graficznym środowiskiem programowania LEGO MINDSTORMS EV3, tłumaczymy powiązanie z wcześniejszą wiedzą dotyczącą instrukcji warunkowych, pętli. Uczymy gimnazjalistów na przykładzie prostego robota jak włączać silniki, skręcać, powtarzać czynności, reagować na czujniki. Przykładowe programy do realizacji na zajęciach:

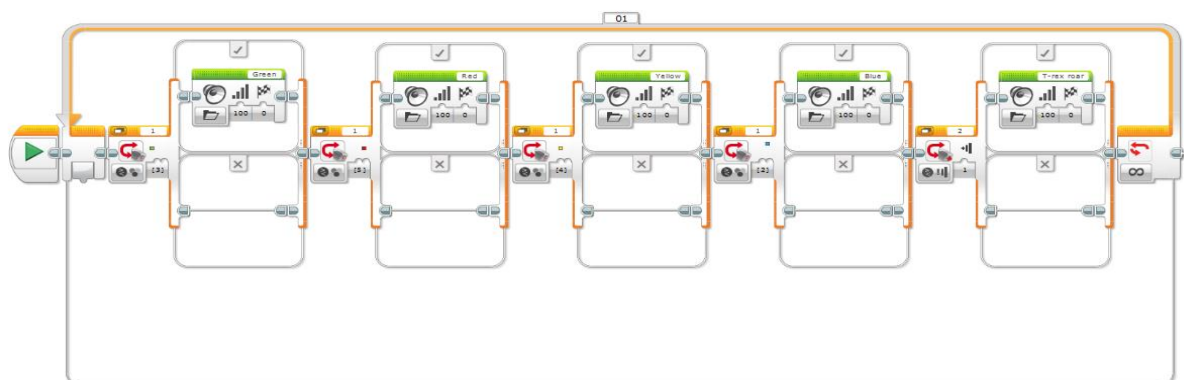
Program – robot jedzie po kwadracie – włącza silniki na 3 sekundy, skręca, powtarza czynność 4 razy



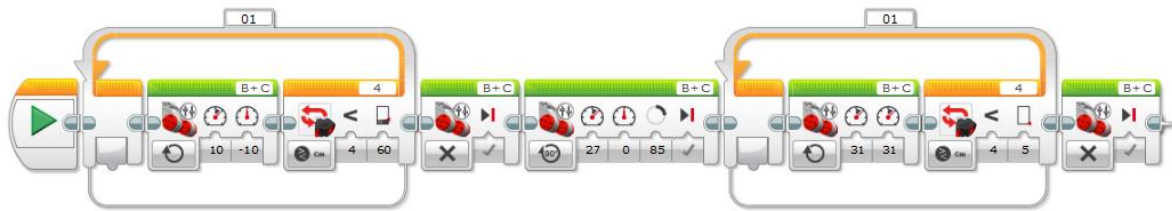
Robot jedzie reagując na przeszkodę przy pomocy czujnika odległości



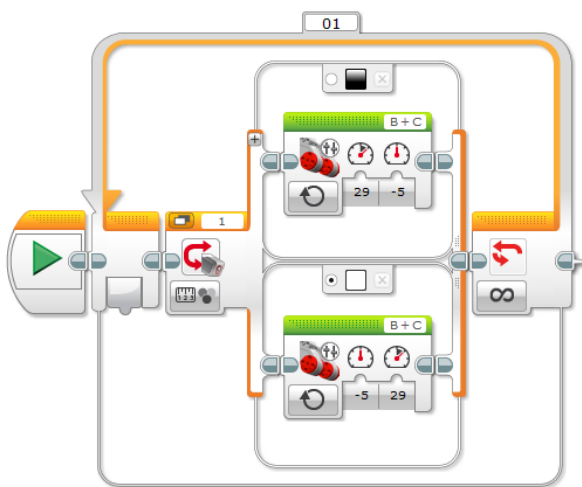
Program reagujący na kolory – wykorzystujemy czujnik, pokazujemy jak działa instrukcja warunkowa



Program wyszukujący element np. puszkę w odległości mniejszej niż 60 cm, korygujący położenie i podjeżdżający do puszki na odległość 3 cm.



Prosty program wykorzystujący pętle i instrukcje warunkową, robot jeździ po czarnej linii.



Realizacja algorytmów przy pomocy oprogramowania LEGO jest dla uczniów dość prosta i intuicyjna dzięki środowisku graficznemu. Efekt programu można łatwo sprawdzić po zgraniu i uruchomieniu robota.

3.13. Język C++ - podstawy

Zapoznajemy uczniów ze środowiskiem kompilatora, wprowadzamy instrukcję cout, tłumaczymy pojęcie zmiennej, wprowadzamy dane przy pomocy instrukcji cin. Realizujemy wybrane algorytmy z części poświęconej schematom blokowym np. objętość prostopadłościanu po wprowadzeniu długości boków a,b,c. Kolejnym krokiem jest instrukcja warunkowa if else oraz instrukcje pętli.

Przykładem może być program na dzielenie:

```
float a,b,c;
cout << „podaj liczbę”
cin >> a;
cout << „podaj kolejną liczbę”;
cin >> b;
if(b==0)
{
    cout << „nie wolno dzielić przez zero !”;
}
else
{
    c=a/b;
    cout << wynik dzielenia to c;
}
```

W tej części realizacji programu trzeba cały czas zwracać uwagę na korelację między schematami blokowymi, graficznym programowaniem LEGO oraz językiem C++.

3.14. Język C++ - programowanie zaawansowane

Wprowadzamy zmienną tablicową, wyszukujemy element w tablicy przy pomocy pętli for, wykonujemy operacje na plikach tekstowych – odczytujemy i zapisujemy dane z plików txt. Wykonujemy ćwiczenia na łańcuchach znaków z wykorzystaniem kodowania ASCII np. sprawdzając długość łańcucha, zmieniając wszystkie litery na małe, sprawdzając czy słowo jest anagramem lub palindromem. Na koniec zaawansowanego programowania w C++ warto, aby uczniowie spróbowali swoich sił w projekcie np. gra w statki z wykorzystaniem tablicy 2 wymiarowej.

Przykładowy kod C++ - gra w statki

```
srand(time(NULL));
int t[12][12],w,k,i,z;
// zeruje tablice
for(z=0;z<12;z++)
{

for(i=0;i<12;i++)
{
    t[z][i]=0;
}
}
// losuje 1 masztowca
w=rand()%10+1;
k=rand()%10+1;
t[w][k]=1;
int s;
```

```

// losuje kolejny
for(i=0;i<3;i++)
{

s=0;
do {
    w=rand()%10+1;
    k=rand()%10+1;
    s=t[w][k]+t[w+1][k+1]+t[w][k+1]+t[w-1][k+1]+
    t[w-1][k]+t[w-1][k-1]+t[w][k-1]+t[w+1][k-1]+t[w+1][k];
    }
while(s!=0);
t[w][k]=1;
}

```

```

// 2 masztowiec
for(i=0;i<3;i++)
{
s=0;
do {
    w=rand()%10+1;
    k=rand()%9+1;
    s=t[w][k]+t[w+1][k+1]+t[w][k+1]+t[w-1][k+1]+
    t[w-1][k]+t[w-1][k-1]+t[w][k-1]+t[w+1][k-1]+t[w+1][k]
    +t[w][k+2]+t[w-1][k+2]+t[w+1][k+2];
    }
while(s!=0);
t[w][k]=1;
t[w][k+1]=1;
}

```

```

// 3 masztowiec
for(i=0;i<2;i++)
{
s=0;
do {
    w=rand()%10+1;
    k=rand()%8+1;
    s=t[w][k]+t[w+1][k+1]+t[w][k+1]+t[w-1][k+1]+
    t[w-1][k]+t[w-1][k-1]+t[w][k-1]+t[w+1][k-1]+t[w+1][k]
    +t[w][k+2]+t[w-1][k+2]+t[w+1][k+2]
    +t[w][k+3]+t[w-1][k+3]+t[w+1][k+3];
    }
while(s!=0);
t[w][k]=1;
t[w][k+1]=1;
t[w][k+2]=1;
}

```

```

// kontrolnie wyswietlam !!!! BEZ RAMKI !!!!
// tu litery A B C
cout << " ABCDEFGHJ" << endl;
for(z=1;z<=10;z++)
{
if(z<10)
cout << z << " ";
else
cout << z << " ";
for(i=1;i<=10;i++)
{
cout << t[z][i];
} cout << endl;
}

```

3.15. C++ - wybrane algorytmy klasyczne

Część zajęć w czasie których uczniowie poznają wybrane algorytmy np. największy wspólny dzielnik, sortowanie bąbelkowe, pierwiastek kwadratowy z liczby, sito Eratostenesa. Warto uczniów zapoznać z listą kroków i zasadą działania algorytmu i polecić napisanie kodu w C++. Rozbudowanie tej części realizacji programu można modyfikować w zależności od tempa pracy, realizacji wcześniejszych treści oraz zainteresowania uczniów programowaniem.

4. Przewidywane osiągnięcia uczniów

Po zakończeniu realizacji programu uczeń powinien:

- Sprawnie posługiwać się systemem operacyjnym,
- Znać budowę komputera oraz zasadę działania urządzeń zewnętrznych,
- Dostrzegać zalety i niebezpieczeństwa pracy sieci komputerowych,
- Sprawnie wyszukiwać i przetwarzać informację,
- Korzystając z edytora grafiki wektorowej tworzyć kompozycję z figur,
- Stosować edytor grafiki rastrowej,
- Znać zasady obowiązujące przy redagowaniu tekstów w edytorze komputerowym,
- Dostrzegać korzyści płynące z używania arkusza kalkulacyjnego do obliczeń i tworzenia wykresów,
- Stosować ze zrozumieniem metody wyszukiwania informacji w bazach danych,
- Tworzyć i przedstawiać prezentację na wybrany temat,
- Komunikować się w sieci w wykorzystaniu netykiety,
- Stworzyć stronę internetową z wykorzystaniem HTML i CSS,

- Wyjaśniać pojęcie algorytmu,
- Znać budowę modułową komputera,
- Formułować ścisły opis sytuacji problemowej,
- Rozwiązywać wybrane zadania algorytmiczne na schematach blokowych i oprogramowaniu LEGO,
- Tworzyć programy i je testować przy pomocy robotów LEGO,
- Programować wybrane czujniki LEGO Mindstorm,
- Pisać proste programy w języku C++,
- Znać funkcje daty, czasu oraz tekstowe w arkuszu kalkulacyjnym,
- Tworzyć zaawansowane animacje komputerowe np. w programie Flash,
- Modelować symulację zjawisk np. fizycznych czy chemicznych,
- Pisać zaawansowane programy komputerowe (pętle, tablice, zapis odczyt z pliku, struktury danych)
- Rozwiązywać zaawansowane problemy algorytmiczne wykorzystując roboty LEGO i C++
- Znać i opisywać wybrane algorytmy klasyczne
- Charakteryzować algorytmy sortowania danych
- Przygotowywać wybrane zestawienia danych
- Wykorzystywać programy komputerowe wzbogacające i wspomagające naukę w wybranych dziedzinach
- Wymieniać zagadania etyczne i prawne związane z ochroną własności intelektualnej
- Znać przejawy przestępczości komputerowej i przepisy prawne z tym związane