| | | | |
|---|---|---|---|
| Numer: | **MATINF 4/2023** | Stron: | 19 |
| Data wydania: | 20 listopada 2023 | Druk: | bez drukowanych egzemplarzy |
| Adresy Redakcji: | czasopismo.matinf@gmail.com | Witryny informacyjne, regulaminy: | https://github.com/czasopismo-MATINF/czasopismo-MATINF |
| Czytelnik: | osoba samodzielnie ucząca się, student | | |
| Cel: | systematyczne kursy podstaw programowania, matematyki, algebry i analizy matematycznej, artykuły o takowej tematyce; recenzje książek do nauki, narzędzi do programowania, listy zasobów informatycznych, serwisów internetowych; po dwudziestu latach wybuchu programistycznego niepotrzebna już promocja programowania, ale powrót na jego podstawowy poziom i znalezienie miejsca dla jego podstaw przy potwornie zwiększającym się stopniu skomplikowania algorytmów oraz narzędzi programistycznych; systematyczny zbiór powszechnie znanych pomysłów i idei; | | |


## Wstęp do bieżącego numeru:

W tym numerze grudniowym kolejne sprawozdania z kursów programowania poprzez języki Python oraz Java, szkice z kursów analizy matematycznej oraz algebry. Ponadto: o robotowych pomocach do nauki programowania, teleskopach daleko od Ziemii, próbie analizy azjatyckiego rynku elektroniki oraz kolejny odcinek leksykonu.

## Spis aktualnie rozwijanej zawartości:

# Szkic 2. O narzędziach do programowania oraz nauki programowania w starych i nowych typach języków.

Redakcja postanowiła wyobrażić możliwe ulepszenia (?) procesu kodowania z ::MATINF-2/2023::. Redakcja jest przyzwyczajona do ręcznych długopisów, zatem przed leży duży płat (telewizor, ekran, monitor) elektroniczny (?), po którym pisze. Wokół biurka leżą roboty - zmienne. Wskazując długopisem na robota, następnie, pisząc znak na ekranie, tworzy jego symbol. Dotykając symbolu robota oraz podając na głos komendy wydaje polecenia. W 2023 roku są już firmy, które próbują stworzyć połączenie mózg - komputer (brain-computer interface) (!): czy na zasadzie intefejsu (interface) - skończonego zestawu funkcji-komunikatów?

Czy klawiatura ze znakami o zmiennej konfiguracji w formie składanych przycisków jest potrzebna? Czy program powinien być zapisany w pamięci trwałej? Czy w ogóle wypowiedzi języków mówionych oraz migowych dają się zapisać w formie dyskretnego ciągu znaków:

współbieżny program jazdy na rowerze po kwadracie o boku długości dziesięciu metrów: [ [L>] [P>] ] ... || [ -10m-[=>] [=] ] ...

Czy roboty wypełnione AI będą uczyć się naśladując zwierzęta, dzieci oraz człowieka (!) ?

Podstawy mechaniki oraz AI z początku XXI wieku sugerują takie rozwiązanie. Mimo to brakuje podstawowych narzędzi: n.p. przenoszenie fragmentów kodu z jednego ekranu na drugi. Przez dwadzieścia lat rozwoju sprzętu komputerowego Redakcja nie widziała pomysłu takiego kopiującego długopisu zrealizowanego. W dostępnym sprzęcie trudno jest zmieniać konfigurację: brakuje zestawów uniwersalnych części do budowania robotów lub generatorów takich części (?), chociaż modele klocków oraz drukarki 3d modeli są dostępne.

Nauczając programowania, Redakcji przydałby się duży ekran, klawiatura z dużą ilością konfigurowalnych piktograficznych znaków, lokalna sieć z podłączonymi robotami (najłatwiej i najprzyjemniej byłyby łódki na osiedlowej kałuży, zalewie lub stawie) oraz zestaw modeli przedstawiających ułożenie robotów, które automatycznie odwzorowują. Czy to samochodziki, czy łódki, czy drony ... Do 2023 roku nauczanie programowania nie przeszło jeszcze powszechnie granicy program - rzeczywistość. Okulary, hełmy do wizyt w rzeczywistości 3d oraz kontrolery 3d również nie są powszechnie w użyciu podczas nauki programowania.

Jeśli ciężko je skonstruować, to chyba właśnie z powodu braku zestawialnych lub ich generatorów części do budowania robotów. A po podniesieniu maski współczesnego samochodu, już wygląda jak zbudowany z klocków.

# Kurs programowania poprzez język Python.

## Sprawozdanie 4.

Po wstępnym porządkowaniu i testowaniu ściągawki Redakcja rozumie lepiej podstawowe kolekcje języka Python. Udało się je wykorzystać w tworzeniu mapy, po której porusza się robot, w skrypcie mapa_robot_i_przeszkody, ale czy do stycznia zamienić klasy na tuple, Redakcja zastanowi się. Tuple nie są bezpośrednio nazywalne, ani ich składowe, są niemodyfikowalne oraz, zdaniem Redakcji, wstawia się je bezpośrednio w kodzie, chociaż można je kopiować. Dużo programując w Javie, Redakcja jest przyzwyczajona do paradygmatu klasowo-obiektowego. Do skryptu z robotem i mapą Redakcja dodała geometryczne, w większości czarno-białe efekty pułapek. Redakcja prawie ukończyła testowanie i porządkowanie ściągawki.

W kursie z użyciem języka Java, Redakcja korzysta z OOP (object-oriented programming) oraz OOD (object-oriented design), a w tym postanowiła przyjrzeć się programowaniu funkcyjnemu.

Język Python ma bardzo ładnie zbudowane przekazywanie argumentów (arguments, actual parameters) do wywołań (call) funkcji. Argumenty to wartości (values), są podstawiane za parametry formalne (formal parameters), które mogą być obowiązkowe (mandatory) oraz nieobowiązkowe (optional) z wartościami domyślnymi (defaults, default values). Argumenty są przekazywane przez wartość jako referencje obiektów (!): x = [...] \ func(x) : \ x = {...} - zewnętrzny x oraz wewnętrzny x są referencjami do tej samej listy, ale zmiana wewnętrznego nie zmienia zewnętrznego. Zdaniem Redakcji, jeszcze nie doczytała, funkcje w Pythonie muszą być obiektami (function objects), ponieważ wartości domyślne parametrów są obliczane raz, chyba przy pierwszym wywołaniu funkcji, zatem muszą być zapisane n.p. razem z obiektem funkcji przechowującym jej kod, a zatem, zdaniem Redakcji, funkcje w języku Python będzie można przekazywać jako argumenty do innych funkcji. Redakcja tutaj zgaduje, ale świadczą o tym również wyrażenia lambda (lambdas, lambda expressions) wbudowane w język Python. Zwykle, obie te konstrukcje w językach programowania występują jednocześnie. Zatem, przynajmniej częściowo, język Python spełnia paradygmat programowania funkcyjnego.

W języku Python można przekazać dodatkowe argumenty (variadic arguments, arbitrary number of arguments) spoza listy potrzebnych funkcji (przeciążanie, method overloading w Javie) przy pomocy parametrów oznaczonych "*" (tuple lub lista ?) oraz "**" (słownik). Parametry tylko pozycyjne (positional-only) można zostawić po lewej, oddzielone znakiem "/", a po znaku "*", po prawej oddzielić parametry przekazywane tylko z nazwami (keyword arguments, keyword-only).

/, *list, *, **dict

Argumenty przekazywane w liście lub słowniku można "rozpakować" (unpacking) przy pomocy operatorów (?) range oraz **.

Czym są adnotacje oraz komentarze dokumentujące nie do końca widać z przykładów. Adnotacje, podobnie w języku Java, mogą służyć od tworzenia narzędzi, które wspomagają programowanie, podobnie komentarze dokumentujące: n.p. można do nich wstawiać formuły w innych językach oznaczające przetwarzanie kodu -

n.p. """ ::JAVA:: for(int i = 0; i < 10; ++i) print(i); """ byłaby kodem, który po automatycznym przetłumaczeniu przez odpowiednie narzędzie na język Python mógłby być wstawiany do funkcji (kodowanie w kilku językach).

częściowe podsumowanie linków zbieranych podczas czytania kursu:

https://docs.python.org/3/tutorial/
https://docs.python.org/3/library/index.html
https://docs.python.org/3/reference/index.html
https://docs.python.org/3/glossary.html
https://peps.python.org/pep-0008/

https://docs.python.org/3/using/index.html
https://docs.python.org/3/faq/programming.html
https://docs.python.org/3/contents.html

https://pypi.org/

https://peps.python.org/pep-3107/
https://peps.python.org/pep-0484/

# Kurs programowania poprzez język Java.

## Sprawozdanie 4.

Po przetłumaczeniu z języka Python na jezyk Java, nie bez problemów spowodowanych dynamicznym w Pythonie oraz statycznym w Javie typowaniami zmiennych, Redakcja postanowiła przemyśleć organizację oraz rozmieszczenie atrybutów w klasach, chociaż małego, skryptu mapa_robot_i_przeszkody, czyli: OOD (object-oriented design). W takim celu najlepiej korzystać z dużej, białej, suchościeralnej tablicy na alkoholowe flamastry, na której rysowane w graficznych wersjach języka UML (Unified Modeling Language) pomysły można łatwo modyfikować: szybko, na bieżąco projektować, przerabiać, ścierać oraz przenosić, czy bardziej rozdzielać czy łączyć funkcjonalności. W trakcie takiego projektowania, nawet przy niewielkich projektach, bardzo często odnajduje się niespodziewane podobieństwa oraz różnice, które upraszczają przyszłe zmiany oraz zarządzanie programem oraz kodem. Dopiero wiele później, po poznaniu programowania OOP (object-oriented programming) Redakcja zaczęła rozrysowywać tworzone programy, wcześniej uznając, że proste łatwo uchwycić. Jaką jest to podporą, skarbnicą, kopalnią rozwiązań i pomysłów, pomocą w ogarnięciu, porządkowaniu i sprzątaniu programu, poznaje się bardzo szybko. Redakcja opisze i wprowadzi zmiany przed następnym sprawozdaniem.

W załączonej liście linków znajdują się adresy z odpowiadającymi stronami czytanego kursu o OOP. Zamiast tworzyć kolejny w internecie opis pojęć klas i obiektów, przed następnym sprawozdaniem, Redakcja przygotuje leksykon z użyciem treści z lekcji "Classes and Objects" oraz "Interfaces and Inheritance" umieszczonych na stronach firmy Oracle Corporation.

częściowe podsumowanie linków zbieranych podczas czytania kursu oraz materiału, którego będzie dotyczyć:

https://docs.oracle.com/javase/8/
https://docs.oracle.com/javase/tutorial/
tutorialLearningPaths.html

https://docs.oracle.com/javase/tutorial/java/index.html

https://docs.oracle.com/javase/tutorial/java/javaOO/
index.html
https://docs.oracle.com/javase/tutorial/java/IandI/
index.html

https://docs.oracle.com/javase/tutorial/essential/
index.html

https://docs.oracle.com/javase/tutorial/collections/
index.html
https://docs.oracle.com/javase/tutorial/java/javaOO/
lambdaexpressions.html
https://docs.oracle.com/javase/tutorial/collections/
streams/index.html

https://docs.oracle.com/javase/tutorial/extra/generics/
index.html

# Sprawozdanie 1. Teleskopy w przestrzeni kosmicznej.

Redakcja, podobnie do początków XXI wieku, postanowiła poczytać o odkryciach astronomicznych a rozpoczęła od hasła "teleskopy kosmiczne. Po otworzeniu kilku linków wyników wyszukiwarki internetowej, wybrała artykuł z czasopisma "Młody Technik" oraz listę takich teleskopów na stronie z Wikipedii. Treści stron z Wikipedii Redakcja nie przeczytała, który to sposób nie nadawał się do zapamiętywania informacji. Artykuł, natomiast, zawiera krótkie notatki dotyczące największych (czy nie interesować się najmniejszymi ?), najbardziej znanych teleskopów w przestrzeni kosmicznej na orbitach wokół Ziemi oraz orbitach wokół Słońca Ziemi!

Po przetłumaczeniu w jednej z przeglądarek internetowych i po przeglądaniu zdjęć z teleskopu światła widzialnego Hubble'a (Edwin Powell Hubble) na stronach internetowych NASA, Redakcja znalazła o wiele pełniejsze zestawienie na stronie handwiki.org zawierające zdjęcia teleskopów, daty trwania misji, typy orbit oraz nie tylko pojedyncze, ale również konstelacje instrumentów teleskopowych oraz plany przyszłych misji. Strona dała się łatwo przetłumaczyć na język polski.

Wygląda na to, że w roku 2023 astronomowie nie mają bliskich planów umieszczenia teleskopów na orbitach "około-układo-słonecznych": orbitach poza orbitą Plutona, poza naszym Układem Słonecznym, chociaż były i są już teleskopy na Księżycu Ziemi oraz orbitach heliocentrycznych. Ani dokładną definicją, ani analizą matematyczną punktów

Lagrange'a (Joseph Louis Lagrange) Redakcja przy tym czytaniu się nie zajmowała.

Przeglądaniem strony o teleskopie Kepler'a (Johannes Kepler) Redakcja się nie zajęła, ponieważ na żadnej z odkrytych planet nie potwierdzono jeszcze istnienia znanego nam z Ziemi biologicznego życia.

Misji jest dużo, a Redakcję nie interesują tyle, by czytać o wszystkich. Po przejrzeniu teleskopów światła widzialnego, Redakcja sprawdziła te, które zajmują się obserwacjami światła podczerwonego (teleskopy mogą być wielofunkcyjne, z wieloma przyrządami), z których największy z wszystkich do tej pory zbudowanych - teleskop Webb'a (James Edwin Webb), przez trzy agencje CSA, NASA, ESA, znajduje się dość daleko od Ziemi. Wygląda na to, że nawet ten, o największej rozdzielczości, może wykryć, ale nie widzi szczegółów powierzchni planet w innych układach solarnych, chociaż może wykryć pojedyncze gwiazdy w nieodległych galaktykach. Redakcja może się mylić. Z dwóch starszych teleskopów światła podczerwonego Spitzer'a (Lyman Spitzer Jr.) oraz Herschel'a (Friedrich Wilhelm Herschel) przestano korzystać. Podobnie, z dwóch satelitów do badania światła mikrofalowego i radiowego (!), rozmiarami w pokoju, które umieszczono daleko od Ziemi: WMAP oraz Planck (Max Karl Ernst, Ludwig Planck), już po pierwszym na orbicie Ziemii - COBE.

Satelitów z urządzeniami obserwującymi w zakresie fal/cząstek wysokoenergetycznych (gorących, jasnych ?): ultrafioletowego, gamma, rentgena (Wilhelm Conrad Röntgen) najwięcej jest na orbicie Ziemi. Cztery poza zainteresowały Redakcję:

Gamma-Ray Burst Polarimeter (GAP), JAXA, orbita heliocentryczna
Neutron Star Interior Composition Explorer (NICER), NASA, Międzynarodowa Stacja Kosmiczna
Spektr-RG, Max Planck Institute for Extraterrestrial Physics, RSRI (Russian Space Research Institute), punkt Lagrange'a L2 układu Ziemia-Słońce
Lunar-based ultraviolet telescope (LUT), CNSA, Księżyc, niewielki element chińskiego programu księżycowego

https://global.jaxa.jp/countdown/f17/overview/ikaros_e.html
https://heasarc.gsfc.nasa.gov/docs/nicer/
https://iki.cosmos.ru/research/missions/spektr-rg
https://en.wikipedia.org/wiki/Chang%27e_3
http://english.nao.cas.cn/Research2015/rp2015/201701/t20170120_173602.html

Cząstki oraz fale grawitacyjne:

Zgodnie z listą, wszystkie pozostałe detektory cząstek znajdują się na orbicie Ziemi; detektor fal grawitacyjnych ESA Lisa Pathfinder na orbicie heliocentrycznej.

https://www.elisascience.org/
https://sci.esa.int/web/lisa-pathfinder/

Czas dostępu do teleskopu, prawdopodobnie, jest przydzielany przez internet naukowcom (research groups) podobnie do przydzielania czasu i miejsca w kawiarniach internetowych. Być może dlatego, że dane z obserwacji są rozproszone, Redakcja nie znalazła jednej strony ze wszystkimi surowymi danymi z obserwacji teleskopów do własnego przetwarzania. Redakcja wyobraża, że wysłanie przyrządu na orbitę można zamówić przez internet, podobnie czas komunikacji z przyrządem poprzez naziemne stacje z antenami, ponieważ internet miedzyplanetarny jeszcze nie działa. Na dole strony znajdują się listy innych obiektów astronomii i kosmologii.

https://handwiki.org/wiki/Astronomy:List_of_space_telescopes

https://pl.wikipedia.org/wiki/Teleskop_kosmiczny

https://mlodytechnik.pl/technika/30012-teleskopy-i-obserwatoria-kosmiczne
https://mlodytechnik.pl/i/images/4/4/3/dz0zNTQzJmg9MjA2Nw==_src_17443-2-Porownanie-najbardziej-znanych-teleskopow-kosmicznych.jpg

https://www.nasa.gov/mission_pages/hubble/main/index.html
https://hubblesite.org/ (jedyne zdjęcia na tej stronie, które były interesujące dla Redakcji, to "Hubble Favorites"; łatwa w użyciu, nieprofesjonalna wyszukiwarka zdjęć)

https://www.nasa.gov/mission_pages/kepler/main/index.html
https://exoplanets.nasa.gov/
https://www.nasa.gov/astrobiology/

https://webb.nasa.gov/
https://www.flickr.com/people/nasawebbtelescope/
https://esawebb.org/
https://www.asc-csa.gc.ca/eng/satellites/jwst/
https://www.asc-csa.gc.ca/eng/satellites/jwst/science-images-from-james-webb-space-telescope.asp

https://science.nasa.gov/mission/spitzer/
https://www.spitzer.caltech.edu/
https://herschel.jpl.nasa.gov/

inne:
https://www.stsci.edu/
https://archive.stsci.edu/
https://nssdc.gsfc.nasa.gov/nmc/SpacecraftQuery.jsp (wyszukiwarka statków kosmicznych)

https://science.nasa.gov/ (menu w lewym, górnym rogu z klasyfikacją)

wspomniane w sprawozdaniu teleskopy:
Hubble'a (Edwin Powell Hubble)
Kepler'a (Johannes Kepler)
Webb'a (James Edwin Webb)
Spitzer'a (Lyman Spitzer Jr.)
Herschel'a (Friedrich Wilhelm Herschel)
COBE, WMAP, Planck (Max Karl Ernst Ludwig Planck)
Gamma-Ray Burst Polarimeter (GAP), Neutron Star Interior Composition Explorer (NICER), Spektr-RG, Lunar-based ultraviolet telescope (LUT)
Lisa Pathfinder
Euclid (

Redakcji próba klasyfikacji kategorii obiektów oraz zdarzeń kosmicznych, na które kierowane są teleskopy kosmiczne:
ciemna materia, ciemna energia, fale grawitacyjne
deep fields
czarne dziury oraz ich dżety, kwazary
galaktyki, gromady galaktyk
mgławice (nebula, nebulae), pył gwiezdny, otoczki gwiazd, echa świetlne
przejścia na tle, zderzenia, kolizje, zjawiska atmosferyczne na gwiazdach oraz planetach
zbiory (gromady, klastry) gwiazd
gwiazdy (zmienne, cefeidy, pulsary, supernowe, "umierające", białe karły, neutronowe, układy podwójne, jest wiele typów, ...)
planety, księżyce, asteroidy, komety
atmosfery zewnętrznych planet oraz księżyców Układu Słonecznego

https://worldwidetelescope.org/ (!)

https://www.urania.edu.pl/tagi/teleskopy-kosmiczne (w całym czasopiśmie informacje są rozrzucone, artykuły za trudne)

Po napisaniu tego sprawozdania Redakcja znalazła informacje o nowym na orbicie "halo" wokół punktu L2 Lagrange'a układu Ziemia-Słońce - teleskopie światła widzialnego Euclid.

https://www.euclid-ec.org/

## Sprawozdanie 2. O nieudanej próbie analizy azjatyckiego rynku elektroniki.

Redakcja postanowiła przygotować graf powiązań pomiędzy firmami produkującymi elektronikę - ograniczając się do wschodniej Azji (Taiwan, Chiny, Korea, Japonia). Po skorzystaniu z wyszukiwarki internetowej, by znaleźć największe, okazało się, że na stronach internetowych listy wybierają firmy według różnych cech - dochody, przychody, kapitalizacja rynku, ... - oraz są podzielone na roczniki. Najpierw Redakcja próbowała segregować firmy zapisując linki w przeglądarce oraz czytając o nich krótkie akapity, potem, by mniej używać myszki - ręcznie, ołówkiem w zeszycie. Po otworzeniu znalezionej listy ze 100 największymi, tylko chińskimi firmami elektronicznymi Redakcja - zrezygnowała - wstępnie ... Po odpoczynku, w kilku sesjach, wybierając po jednej liście i sprawdzając informacje na stronach Wikipedii, stworzeniu własnej, prostej klasyfikacji nieodróżniającej części od chyba prostszych półprzewodników oraz grupując przetwarzanie obrazu razem z produkcją kamer, aparatów, optyki, Redakcja czytała listy do momentu, w którym przeglądanych firm zaczęła rozpoznawać nazwy. W następnych sesjach Redakcja sprawdzała uprzednie, szybko przygotowane dane. Przez półprzewodników, zdaniem Redakcji, należy rozumieć produkcję procesorów, układów scalonych, chip-ów.

Redakcji nie udało się najważniejsze: powiazać firmy w grupy własnościowe oraz w zależności, kto komu zleca produkcję części. Również, Redakcja zrezygnowała z przygotowania mapy fabryk, których firmy mogą operować po kilkaset.

Redakcja zastanawia się, jak można sprawdzić, czy zamawiany w dużych ilościach sprzęt będzie spełniał wymagania zakupu. Zdaniem Redakcji, najlepiej rozkręcać stary sprzęt elektroniczny, uczyć się występujących tam części oraz standardów komunikacji pomiędzy nimi. Klasyfikacja Redakcji dzieli według produktów: półprzewodniki, części elektroniczne, podzespoły, gotowa elektronika użytkowa, elektronika przemysłowa, tylko montaż. Być może, niektóre firmy dodają tylko markę do zakupionych urządzeń. Innym sposobem byłoby prześledzenie wieloetapowej, wielomiejscowej linii produkcyjnej urządzeń (smartfony, laptopy, elektroniczny sprzęt kuchenny, łazienkowy, biurowy).

W jaki sposób w skomplikowanym grafie powiązań zarządza się zgodnością techniczną części oraz oprogramowania?

Redakcja wyobraża:

IDE do projektowania części elektronicznych oraz podzespołów generujące pliki opisu sprzętu HDL, podobnie oprogramowania - SDL, wysyłane do fabryki łącznie (HaSDL), z nadzorem osobistym produkcji przez firmę zakupującą.

Dlaczego w ten sposób nie opracować standardów klocków do konstrukcji robotów: macierzy-szkieletów z gniazdami do podłączeń (podobnie do złącz dysków twardych, gniazd jack, usb, hdmi i.t.p.) z mocowaniem mechanicznym, standardów komunikacji oraz zasilania, zestawów kompatyblinych ramion z chwytakami, szczypcami, kołami, ... ?

Automatycznie generowany system operacyjny, automatycznie generowany sklep oprogramowania, aktualizacje - taki zestaw pozawalałby na łatwe przejście - również organizacyjne - pomiędzy zmianami architektur sprzętu, oprgoramowania, sieci: projektowanie oraz produkcja w starej architekturze a produkcja nowej.

https://en.wikipedia.org/wiki/List_of_largest_manufacturing_companies_by_revenue
https://pl.wikipedia.org/wiki/Najwi%C4%99ksze_przedsi%C4%99biorstwa_elektroniczne_w_kolejnych_latach
https://companiesmarketcap.com/electronics/largest-electronic-manufacturing-by-market-cap/

https://inzynieria.com/artykul/szczegoly/25390,50-najwiekszych-firm-sektora-it-i-elektroniki-uzytkowej
(wiadomości oraz zdjęcia aktualnych konstrukcji - mosty, wiadukty, trasy - inżynieria lądowa, wodna)

ponad 600 firm (tej Redakcja już ręcznie nie przeglądała, chociaż szybko pojawiają się na niej firmy, których nie ma w przygotowanym zestawieniu):

https://disfold.com/industry/electronic-components/companies/ (wyszukiwarka z wszystkich typów branż, ale chyba niepełna:
https://disfold.com/poland/industry/airlines/ (jedna polska linia lotnicza))

100 chińskich firm:

https://www.edn.com/electronic-business-top-100-chinese-electronics-companies/

https://toplist.info/
https://firmsworld.com/

https://www.chinadaily.com.cn/
https://www.scmp.com/

https://tek.info.pl/article/2310/najwieksze_firmy_elektroniczne_w_polsce_2022/18

https://www.eventseye.com/fairs/t1_trade-shows_electronics-electrotechnics.html

https://en.wikipedia.org/wiki/Semiconductor_device_fabrication
https://en.wikipedia.org/wiki/List_of_integrated_circuit_manufacturers
https://en.wikipedia.org/wiki/List_of_semiconductor_fabrication_plants

Przyporządkowanie może zawierać dużo błędów. Już po dodatkowych sesjach z Wikipedią, dokładniejszym sprawdzaniu wcześniej przygotowanych danych, Redakcji przygotowany podział okazał się bardzo niedokładny.

/*****//*****//*****//*****/

Produkcja półprzewodników, części, sprzęt docelowy:

Hon Hai Precision Industry Co., Ltd., Foxconn, Taiwan, https://en.wikipedia.org/wiki/Foxconn

Huawei Technologies Co., Ltd., Chiny, https://en.wikipedia.org/wiki/Huawei
Lenovo Group Limited, Chiny, https://en.wikipedia.org/wiki/Lenovo
Xiaomi, Chiny, https://en.wikipedia.org/wiki/Xiaomi

LG Corporation (LG Group), Korea Południowa, https://en.wikipedia.org/wiki/LG (elektronika, chemia, energia)
Samsung Electronics Co., Ltd., Korea Południowa, https://en.wikipedia.org/wiki/Samsung_Electronics

Fujitsu Limited, Japonia, https://en.wikipedia.org/wiki/Fujitsu (oprogramowanie, usługi, jedna z najstarszych)
Hitachi, Ltd., grupa, Japonia, https://en.wikipedia.org/wiki/Hitachi (dyski, sprzęt budowlany, wojskowy, bardzo różnorodna oferta, jedna z najstarszych)
Mitsubishi Group : Mitsubishi Electric Corporation, keiretsu, Japonia, https://en.wikipedia.org/wiki/Mitsubishi_Electric (lotnictwo, samochody, zakłady zbrojeniowe, chemiczne, huty, stocznie, bank, jedna z najstarszych)
NEC Corporation (członek Sumitomo Group - keiretsu, od XVII wieku), Japonia, https://en.wikipedia.org/wiki/NEC (półprzewodniki teraz w Renesas Electronics)
Panasonic Holdings Corporation, Japonia, https://en.wikipedia.org/wiki/Panasonic (jedna z najstarszych)
Sharp Corporation, Japonia, https://en.wikipedia.org/wiki/Sharp_Corporation (większość posiada Foxconn, jedna z najstarszych, konglomerat)
Sony Group Corporation, Japonia, https://en.wikipedia.org/wiki/Sony (konglomerat)
Toshiba Corporation, Japonia, https://en.wikipedia.org/wiki/Toshiba (konglomerat)

/*****//*****//*****//*****/

Komponenty, części komputerowe:

Compal Electronics, Inc., Taiwan, https://en.wikipedia.org/wiki/Compal_Electronics (produkcja dla innych firm, jedna z największych)
Delta Electronics, Inc., Taiwan, https://en.wikipedia.org/wiki/Delta_Electronics
eMemory Technology Inc., Taiwan
Formosa Plastics Group, Taiwan, https://en.wikipedia.org/wiki/Formosa_Plastics_Group (chemia, paliwa, konglomerat)
Innolux Corporation, Taiwan, https://en.wikipedia.org/wiki/InnoLux_Corporation (produkcja ekranów dla innych firm)
Lite-On, grupa, Taiwan, https://en.wikipedia.org/wiki/Lite-On

MediaTek Inc., Taiwan, https://en.wikipedia.org/wiki/MediaTek
Taiwan Semiconductor Manufacturing Company Limited (TSMC), Taiwan, https://en.wikipedia.org/wiki/TSMC (jedna z największych, produkcja dla innych firm)
Unimicron Technology Corporation, Taiwan, https://en.wikipedia.org/wiki/Unimicron
Yageo Corporation, Taiwan, https://en.wikipedia.org/wiki/Yageo

BOE Technology Group Co., Ltd., Chiny, https://en.wikipedia.org/wiki/BOE_Technology

SK hynix Inc., Korea Południowa, https://en.wikipedia.org/wiki/SK_Hynix (produkcja dla innych firm)

Ibiden Co., Ltd., Japonia, https://en.wikipedia.org/wiki/Ibiden (części samochodowe, jedna z najstarszych)
Micron Memory Japan, K.K., Japonia, https://en.wikipedia.org/wiki/Micron_Memory_Japan (należy do amerykańskiej firmy Micron Technology, Inc.)
Murata Manufacturing Co., Ltd., Japonia, https://en.wikipedia.org/wiki/Murata_Manufacturing (robot jeżdżący na rowerze)
Nichicon Corporation, Japonia, https://en.wikipedia.org/wiki/Nichicon (kondensatory)
Nidec Corporation, Japonia, https://en.wikipedia.org/wiki/Nidec
Renesas Electronics, Japonia, https://en.wikipedia.org/wiki/Renesas_Electronics (półprzewodniki)
Rohm Semiconductor, Japonia, https://en.wikipedia.org/wiki/Rohm (półprzewodniki)
TDK Corporation, Japonia, https://en.wikipedia.org/wiki/TDK (tej samej technologii taśma w starych kasetach magnetofonowych i dyskietkach!)
Taiyo Yuden Co., Ltd., Japonia, https://en.wikipedia.org/wiki/Taiyo_Yuden
Tokyo Electron Limited, Japonia, https://en.wikipedia.org/wiki/Tokyo_Electron
Yaskawa Electric Corporation, Japonia, https://en.wikipedia.org/wiki/Yaskawa_Electric_Corporation (motory elektryczne, jedna z najstarszych)

/*****//*****//*****//*****/

Kamery, aparaty, optyka, grafika, przetwarzanie obrazu, druk, gry:

Largan Precision Company Limited or Largan Precision, Taiwan, https://en.wikipedia.org/wiki/Largan_Precision
VisEra Technology Co., Ltd.

Hikvision, Chiny, https://en.wikipedia.org/wiki/Hikvision

Canon Inc., Japonia, https://en.wikipedia.org/wiki/Canon_Inc.
Konica Minolta Co., Ltd, Japonia, https://en.wikipedia.org/wiki/Konica_Minolta
Nikon Corporation, w grupie Nikon Group, Japonia, https://en.wikipedia.org/wiki/Nikon
Nintendo, Japonia, https://en.wikipedia.org/wiki/Nintendo
Olympus Corporation, Japonia, https://en.wikipedia.org/wiki/Olympus_Corporation (jedna z najstarszych)
Ricoh Company, Ltd., Japonia, https://en.wikipedia.org/wiki/Ricoh

/*****//*****//*****//*****/

Domowe, użytkowe, narzędzia:

Acer Inc., Taiwan, https://en.wikipedia.org/wiki/Acer_Inc.
ASUSTeK Computer Inc., Taiwan, https://en.wikipedia.org/wiki/Asus (również komponenty)

Qisda Corporation, "parent company" of BenQ Corporation, w BenQ Qisda Group, Taiwan, https://en.wikipedia.org/wiki/BenQ (sprzęt medyczny)
Quanta Computer Incorporated, Taiwan, https://en.wikipedia.org/wiki/Quanta_Computer (sprzęt medyczny, kamery, dla innych firm)

BBK Electronics Corporation, Chiny, https://en.wikipedia.org/wiki/BBK_Electronics (ze strony Wikipedii: firma wyrejestrowana ?!)
Gree Electric Appliances Inc. of Zhuhai, Chiny, https://en.wikipedia.org/wiki/Gree_Electric
Hisense Group, Chiny, https://en.wikipedia.org/wiki/Hisense (wiele marek, w tym Toshiba)
Meizu Technology Co., Ltd., Chiny, https://en.wikipedia.org/wiki/Meizu (w większości posiadana przez Zhejiang Geely Holding Group Co., Ltd. (ZGH) - firma samochodowa)
Midea Group, Chiny, https://en.wikipedia.org/wiki/Midea_Group (ze strony Wikipedii: największy na świecie producent robotów)
Techtronic Industries Company Limited, Hong Kong, https://en.wikipedia.org/wiki/Techtronic_Industries (narzędzia)

Omron Corporation, Japonia, https://en.wikipedia.org/wiki/Omron (sprzęt medyczny, pierwsze bramki biletowe, jedna z najstarszych)

/*****//*****//*****//*****/

Składanie (assembly), sprzęt docelowy, dla innych marek:

Wistron Corporation, Taiwan, https://en.wikipedia.org/wiki/Wistron

Pegatron Corporation, Chiny, https://en.wikipedia.org/wiki/Pegatron

SIIX Corporation, Japonia, https://ja.wikipedia.org/wiki/%E3%82%B7%E3%83%BC%E3%82%AF%E3%82%B9

# Leksykon 3. Paradygmaty.

Ze względu na pozorną prostotę a rzeczywistą złożoność Redakcja postanowiła pisać fragmenty powtórek w leksykonie. Internetowy słownik języka polskiego PWN dla hasła "paradygmat", inne źródła:

(filozofia) «przyjęty sposób widzenia rzeczywistości w danej dziedzinie, doktrynie itp. (np.paradygmat

ontologiczny, mentalistyczny, lingwistyczny)»

(językoznawstwo) «zespół form fleksyjnych danego wyrazu lub końcówek właściwych danej grupie wyrazów; wzorzec, model

deklinacyjny lub koniugacyjny»

Pominięte przez Redakcję w "Leksykonie 1." drugie z tych objaśnień ma duże znaczenie w kontekście informatycznym. Ostatnio, przeglądając strony internetowe, Redakcja znalazła proponowaną definicję pojęcia paradygmatu w programowaniu jako narzędzia uwalniającego od klasy błędów programistycznych w przypadku wielu sposobów myślenia "do wyboru", w zależności od rodzaju projektu, jego wymagań, umiejętności programisty, możliwości IDE.

Żadne z części leksykonu na pewno nie będą pełne.

dotyczące instrukcji:
skalarne, wektorowe, macierzowe

Zdaniem Redakcji, analizowanie gramatyki języka jest sposobem na nauczenie się wyrażeń i operatorów.

https://docs.python.org/3/reference/lexical_analysis.html#line-structure
https://docs.python.org/3/reference/expressions.html

Poniżej pojęcia - do wykorzystania jako spis treści - dotyczące analizy leksykalnej, zmiennych oraz wyrażeń, których źródłem są głównie dokumentacje języków (tutaj języka Python), fragmenty gramatyki języka Python:

logical line, physical line
comment, encoding declaration
line joining, indentation, white spaces
tokens : identifiers (names), keywords, literals, operators, delimiters
soft keywords, reserved identifiers
literals (string, numeric, ...)

```
atom       ::=  identifier | literal | enclosure
literal    ::=  stringliteral | bytesliteral | integer | floatnumber | imagnumber

enclosure        ::=  parenth_form | list_display | dict_display | set_display | generator_expression | yield_atom
parenth_form     ::=  "(" [starred_expression] ")"

list_display     ::=  "[" [starred_list | comprehension] "]"
dict_display     ::=  "{" [dict_item_list | dict_comprehension] "}"
set_display      ::=  "{" (starred_list | comprehension) "}"


expression_list  ::=  expression ("," expression)* [","]
starred_list     ::=  starred_item ("," starred_item)* [","]
starred_expression  ::=  expression | (starred_item ",")* [starred_item]
starred_item     ::=  assignment_expression | "*" or_expr

expression                   ::=  conditional_expression | lambda_expr
```

```
conditional_expression ::=  or_test ["if" or_test "else" expression]
lambda_expr ::=  "lambda" [parameter_list] ":" expression

or_test  ::=  and_test | or_test "or" and_test
and_test ::=  not_test | and_test "and" not_test
not_test ::=  comparison | "not" not_test

comparison    ::=  or_expr (comp_operator or_expr)*
comp_operator ::=  "<" | ">" | "==" | ">=" | "<=" | "!=" | "is" ["not"] | ["not"] "in"

assignment_expression ::=  [identifier ":="] expression
and_expr ::=  shift_expr | and_expr "&" shift_expr
xor_expr ::=  and_expr | xor_expr "^" and_expr
or_expr  ::=  xor_expr | or_expr "|" xor_expr


comprehension     ::=  assignment_expression comp_for
comp_iter   ::=  comp_for | comp_if
comp_for    ::=  ["async"] "for" target_list "in" or_test [comp_iter]
comp_if           ::=  "if" or_test [comp_iter]

target_list ::=  target ("," target)* [","]
target            ::=  identifier | "(" [target_list] ")" | "[" [target_list] "]" | attributeref | subscription | slicing | "*" target


dict_item_list     ::=  dict_item ("," dict_item)* [","]
dict_item            ::=  expression ":" expression | "**" or_expr
dict_comprehension ::=  expression ":" expression comp_for


generator_expression ::=  "(" expression comp_for ")"
yield_atom       ::=  "(" yield_expression ")"
```

```
yield_expression ::=  "yield" [expression_list | "from" expression]
await_expr ::=  "await" primary

primary ::=  atom | attributeref | subscription | slicing | call
[...]
```

pojęcia dotyczące zmiennych, przypisań, wyrażeń, operatorów:
deklaracja (declaration), przypisanie (assignment), l-wartość (l-value), r-wartość (r-value), modyfikowalny (modifiable),
priorytety (operator precedence), argumentowość (arity), przeciążanie (overloading), leniwa ewaluacja (lazy evaluation), (?) (short circuit)

https://docs.python.org/3/reference/lexical_analysis.html#line-structure
https://docs.python.org/3/reference/expressions.html

paradygmaty dotyczące budowania oprogramowania, funkcji frameworków, kompilatorów, IDE (lista na pewno nie jest pełna):

programowania aspektowe (aspect), komponentowe (component, modular), generyczne (generic, template)

adnotacje (annotation), refleksja (reflection) - pojęcia z języka Java, umożliwiają tworzenie narzędzi usprawniających programowanie

# Kurs analizy matematycznej.

## Szkic 4. ::GMFI-RIII::

Powtórka materiału polskiej szkoły średniej lat dziewięćdziesiątych w oparciu ::GMFI-RIII-§1:: nie jest ani gruntowną powtórką, ani systematycznym, aksjomatycznym wyprowadzeniem wzorów na pochodne funkcji elementarnych, w tym trygonometrycznych oraz twierdzeń geometrii. Nieco zawiedziony Czytelnik, jeśli potrzebuje oraz ma czas, powinien wspomóc się dużą, białą, suchościeralną tablicą na alkoholowe flamastry oraz podręcznikami, które już przeczytał, a wierzący w różniczkową "teorię wszystkiego" - gapić się na wzory w ::GMFI-RIII-§1-p.97::. Czytelnik, w zeszycie formatu B5, co najmniej stukartkowym, w twardej okładce, może zrobić listę twierdzeń z analizy oraz geometrii, których dowodów nie widział lub nie umie, odkreślając w ten sposób stopień pośredni w przejściu do tematów bardziej zaawansowanych, jednocześnie zachowując możliwość wyprowadzania twierdzeń od aksjomatów.

Przykłady w ::GMFI-RIII-§1-p99:: - podobnie w całym ::GMF:: - dzielą się na cztery zbiory:

1) obliczeniowych, mało ogólnych
2) ciekawych, obliczeniowo-pomysłowych, wartych zanotowania (::GMFI-RIII-§1-p.99-prz.23::, ::GMFI-RIII-§1-p.102::)
3) geometrycznych (::GMFI-RIII-§1-p.99-prz.27,28::)
4) fizycznych (::GMFI-RIII-§1-p.99-prz.29::)

Czytelnik sam oceni tą klasyfikację oraz wrzuci przykłady do tych zbiorów, ale Redakcja koniecznie zaleca notowanie zbioru numer dwa (bez wyprowadzeń: wzór oraz odnośnik do strony), co pozwala na szybkie powtórki, porównania i zestawienia. Przykłady geometryczne oraz fizyczne służą zrozumieniu pojęcia nieskończenie małej, są bardzo ważne (!), ale ich notowanie byłoby utratą czasu na przerysowywanie oraz powielanie ::GMF::, chociaż nie wystarczają, ponieważ należałoby znać dynamikę Newton'a (Sir Isaac Newton) oraz teorię prądu stałego, to są świetną podstawą do wykorzystania na korepetycjach nie z fizyki, ale z matematyki. Do pełnego zrozumienia przykładów z elektromagnetyzmu, zdaniem Redakcji, potrzebne jest obejrzenie wykładów wideo. Redakcja jak ognia unika przykładów o błędach obliczeniowych, które mogą być ważne dla Czytelnika studiującego metody numeryczne, reprezentacje binarne oraz procesory.

Przed wprowadzeniem pojęcia różniczki w ::GMFI-RIII-§2::, do której zrozumienia najlepiej uczęszczać na równoległe wykłady z przestrzeni Banacha (Stefan Banach), podane są trzy rodzaje notacji pochodnych: szkolna "primowa" (Giuseppe Luigi Lagrangia, Joseph Louis de Lagrange), nieskończenie małych (Gottfried Wilhelm Leibniz) oraz operatorowa Cauchy'ego (Augustin Louis, baron Cauchy). Z teorii przestrzeni Banacha, zdaniem Redakcji, wystarczą: zupełność R^n, odwzorowania liniowe, macierze, normy, twierdzenie o równoważności wszystkich R-norm dla przestrzeni skończenie wymiarowych (!), normy odwzorowań liniowych, wieloliniowość, normy odwzorowań wieloliniowych. Obliczanie pochodnych (granic ilorazów różnicowych) należy odróżnić od obliczania różniczek (przybliżających odwzorowań liniowych) przy pomocy operatora małe d, który stawiamy po lewej stronie operandu korzystając z jego własności liniowości oraz działania na iloczynach (dxy = xdy+ydx) i ilorazach (d(x/y)=xd(1/y)+(1/y)dx). No tak, tutaj również przyda się systematyczna teoria różniczek w przestrzeniach Banacha oraz ich działanie na odwzorowaniach liniowych, wieloliniowych i ich zestawieniach. Redakcja najczęściej wykorzystywała regułę różniczkowania odwzorowania dwuliniowego. Znajomość tych pojęć, rodziny twierdzeń o wartościach średnich z ::GMFI-RIII-§3:: oraz swobodne przekształcanie pomiędzy wzorami z ilorazem różnicowym a tymi z odwzorowaniami liniowymi pozwala na niewymuszone czytanie rozdziałów o wielu zmiennych. Redakcja powtarzała na pamięć:

Twierdzenie Fermat'a (Pierre de Fermat)
Twierdzenie Darboux (Jean Gaston Darboux)
Twierdzenie Rolle'a (Michel Rolle)
Twierdzenie (wzór) Lagrange'a (Giuseppe Luigi Lagrangia, Joseph Louis de Lagrange)
Twierdzenie (wzór) Cauchy'ego (Augustin Louis, baron Cauchy)

Zauważyć warunki założeń o różniczkowalności tylko we wnętrzu przedziałów. Wartości średnie pojawiają się przy rozwinięciach w szereg Taylor'a (Brook Taylor) funkcji wielu zmiennych podczas rysowania kwadratów i sześcianów złożonych z samych krawędzi. Te twierdzenia również należy rozrysować.

Ponieważ obliczanie pochodnych prostych funkcji elementarnych jest mechaniczne oraz algebraiczne, Redakcja zaleca wykorzystanie czasu na szukanie głębszych zależności we wzorach ::GMFI-RIII-§4::. Pojawiają się tam wykorzystanie równań różniczkowych, podstawienia. W przypadku kłopotów ze zrozumieniem przyszłych przykładów lub teorii ::GMF::, często rozwiązaniem jest zidentyfikowanie lub wprowadzenie zmiennej niezależnej, która swobodnie przybiera wartości z pewnego rzeczywistego przedziału, a uzależnione od której są zmienne pozostałe oraz ich różniczki (n.p. x' = dx/dt, y' = dy/dt). Może nie każdy przykład jest działającym silnikiem tłokowo-spalinowym lokomotywy, którego w różnych zakresach różne parametry mierzymy, a ćwiczenia polegają na wyznaczaniu bezpośrednich lub uwikłanych zależności pomiędzy zmiennymi, ich pochodnymi, różniczkami, uważając na punkty oraz krawędzie graniczne oraz osobliwe. Pomimo skupienia ::GMF:: na dwóch i trzech wymiarach, w trakcie czytania Czytelnik powinien zastanawiać się nad wykonywaniem obliczeń z większą ilością zmiennych. W ::GMFI-RIII-§2-p.106:: znowu można popracować nad algebrą dzielenia różniczek, a przy nauce różniczek wyższych rzędów w ::GMFI-RIII-§4-p.119:: oraz następnych punktach Redakcji pomaga teoria odwzorowań wieloliniowych oraz rozróżnianie przyrostów dx^2 = dx1*dx2. Podobnie jak f'(x)dx1 oznacza przybliżanie funkcji przy pomocy pochodnej, tak f''(x)dx1dx2 oznacza przybliżanie pierwszej różniczki przy pomocy drugiej. Podsumowując : na podstawie wzorów w ::GMFI-RIII-§1-p.97::, ::GMFI-RIII-§2-p.105,106::, ::GMFI-RIII-§4-p.119-121:: należy spróbować zrozumieć algebrę różniczek - przybliżających odwzorowań liniowych. ::GMFI-RIII-§4-p.122:: : niby nudne, kombinatoryczne różnice skończone, a obliczenie n-tej pochodnej jednym przejściem granicznym.

Wzór Leibniz'a (Gottfried Wilhelm Leibniz) formaliści udowodnią indukcyjnie, łatwiej przymierzyć do trójkąta Pascal'a (Blaise Pascal) albo stosować regułę różniczkowania wiele razy. Redakcja w swoich notatkach zostawiła kilka kartek na interesujące ją zestawy wielomianów specjalnych, które z podstawowego wykładu analizy zwykle okazują się bazą ortogonalną lub nawet ortonormalną pewnego całkowego iloczynu skalarnego. ::GMFI-RIII-§4-p.118-prz.6:: zawiera wyznaczenie wielomianów Legendre'a (Adrien-Marie Legendre), a za podobne obliczenia Redakcja bardzo lubi ::GMF::. Do zestawów wielomianów specjalnych zwykle jest kilka podejść, które w notatkach Redakcja zaleca zbierać z różnych książek:

wielomiany Legendre'a (Adrien-Marie Legendre)
::GMFII-RIX-§4-p.320::
::GMFII-RXII-§4-p.447-prz.8::
::FLRRiC-RIV-p.28::
::FLRRiC-RX-p.20::

W notatkach Redakcji są informacje przynajmniej o wielomianach:

Bernoulli'ego (Jakob I Bernoulli, Daniel Bernoulli), Czebyszew'a (Пафну́тий Льво́вич Чебышёв), funkcjach Bessel'a

(Friedrich Wilhelm Bessel), Hermitte'a (Charles Hermite), Laguerre'a (Edmond Nicolas Laguerre).

Twierdzenie (lub rodzina twierdzeń) Stone'a-Weierstrass'a (Marshall Harvey Stone, Karl Theodor Wilhelm Weierstraß), którego dowód jest zadziwiająco prosty (n.p. ::KMAI::, stwierdza, że każdą funkcję ciągłą na przestrzeni zwartej można aproksymować funkcjami algebry rozdzielającej (m.in. wielomianami). Każdą funkcję ciągłą na odcinku domkniętym można aproksymować wielomianami z dowolną jednostajną dokładnością. Przybliżanie, aproksymacja, interpolacja. ::GMF-RIII-§5:: podaje metody rozwijania funkcji różniczkowalnych w szeregi Taylora (Brook Taylor) - jednomianów o rosnących stopniach (!), ale nie zawsze jest to aproksymacja. Słynna odwrócona kapeluszowa funkcja e^(-1/x^2) ma wszystkie pochodne w każdym punkcie, w tym wszystkie równe zero w zerze, co nie tak trudno udowodnić (Redakcja nie przypomina dokładnie, ale chyba indukcyjnie, korzystając z ilorazów różnicowych oraz reguły de l'Hospitala (Guillaume François Antoine de L'Hôpital)), zatem wyznaczony szereg jest stale równy zero. Czy da się tą funkcję uciąć i przedłużyć tak, by jej granice w nieskończoności były równe zero zachowując zgodność wszystkich pochodnych? Przedłużanie funkcji, obszarów na brzegach.

Jak szukać wielomianowych szeregów aproksymujących? Jakie funkcje da się aproksymować wielomianami jednorodnymi? Jakie funkcje da się aproksymować wielomianami o z góry ograniczonym stopniu?

::GMF-RIII-§5:: wprowadza wzór Taylor'a (Brook Taylor) z czterema resztami, przy (!) różnych założeniach: reszta "o-małe", reszta Lagrange'a (Giuseppe Luigi Lagrangia, Joseph Louis de Lagrange), reszta Schlomilcha-Roche'a (Oscar Xavier Schlömilch, Édouard Albert Roche?) (!), Cauchy'ego (Augustin Louis Cauchy). W ::GMFII-RIX-§4-p.318:: Czytelnik znajdzie wyprowadzenie wzoru z piątą resztą - całkową. Dla uzupełnienia nazwisk: szereg Maclaurin'a (Colin Maclaurin) to rozwinięcie Taylor'a (Brook Taylor) w zerze. Drażniący jest brak uniwersalnej reszty ... Redakcja zaleca również notowanie rozwinięć funkcji w szeregi w osobnej sekcji zeszytu.

Wzory interpolacyjne ::GMF-RIII-§6:: są proste, jeśli ktoś je wytłumaczy.

Interpolacja oraz aproksymacja jednocześnie? Aproksymacja gładka? (grafika komputerowa)

::FLRRiC:: Franciszek Leja "Rachunek różniczkowy i całkowy"
::KMA:: Krzysztof Maurin "Analiza"

# Kurs algebry.

## Szkic 4. ::SLA-RVIII:: Teoria Galois.

W prezentowanej książce, w rozwinięciach teorii Évariste Galois można dostrzec tendencje do twierdzeń łączących różne kategorie: n.p. grup i ciał (Emil Artin, Ernst Eduard Kummer)

::SLA-RVIII-§8:: Wykorzystanie motywu dualności z ::SLA-RI-§11:: do klasyfikacji rozszerzeń abelowych o wykładniku względnie pierwszym z charakterystyką ciała, potem o wykładniku równym charakterystyce ciała, potem, w oparciu o wektory Witt'a (Ernst Witt) z zadań - o wykładniku równym potędze charakterystyki ciała. Do wykazania poprawności definicji tych wektorów, zdaniem Redakcji, nie są potrzebne ani lemat Gaussa (Johann Carl Friedrich Gauß) z ::SLA-RV-§6::, ani własność jednoznaczności rozkładu UFD (unique factorization domain), tylko odpowiednie operowanie na iloczynach oraz sumach nieskończonych. Spróbować wybrać układ algebraicznie niezależnych liczb rzeczywistych xn, t oraz skorzystać z metod analizy iloczynów nieskończonych do obliczeń pochodnej logarytmicznej (prosto opisane w ::FLRRiC::). Redakcja zaleca jednoczesne rozpisanie trójkąta wzorów współrzędnych górnych (złudnych, secondary components, ghost coordinates) w zależności od współrzędnych dolnych oraz wyliczenie kilku początkowych wzorów na sumę oraz iloczyn. Po przejściu do wybranego pierścienia, wzory wyliczające współrzędne dolne z górnych zawierają ułamki - nie można ich bezpośrednio stosować. Zdaniem Redakcji, z pominięciem definicji F oraz V, rozwiązania zadań 23,24 nie są potrzebne do rozwiązania zadań 25,26. Obliczenia na pierścieniu "logarytmicznych" wektorów Witt'a ograniczonych do indeksów potęg wybranej liczby pierwszej p, by łatwiej zrozumieć, lepiej notować z indeksami p^0, p^1, p^2 ..., wycinając pozostałe współrzędne. Poprawność działań, przemienność, łączność, rozdzielność zostają zachowane dzięki temu, że wzory te zależą właśnie tylko od współczynników równych potęgom wybranej liczby pierwszej. Stwierdzenie, że równanie Ps = x ma w domknięciu algebraicznym rozwiązanie dla każdego x, zdaniem Redakcji, można odczytać bezpośrednio z postaci równań Fs-s = x po przeniesieniu na drugą stronę (Fs = x + s) oraz porównaniu sposobu działania F oraz wielomianów wyznaczających sumę wektorów Witt'a s + x na dolnych współrzędnych. Dzięki F będącemu homomorfizmem, Fs1=Fs2 daje F(s1-s2) = 0, co, ponownie, daje s1-s2 = k x k x ... dzięki sposobie działania F na dolnych współrzędnych. Potem pozostaje już tylko ... sprawdzenie stopnia, abelowości rozszerzenia k(s):k oraz udowodnienie odpowiednika twierdzenia Kummer'a. Czy obcinanie współrzędnych w skończonych wektorach Witt'a jest dzieleniem przez ideał?

Zdaniem Redakcji, dzięki rozkładom grup abelowych na sumy proste p-podgrup, można przejść do rozkładów rozszerzeń o dowolnym skończonym stopniu.

Prezentowana obliczeniowa algebra jakby wchodzi na jeszcze wyższy poziom, staje się inżynieryjna, lecz okazuje się nie trudna, chociaż niejasna.

Tutaj Redakcja przejrzała spisy treści sześciu tomów książek "Handbook of Algebra" pod redakcją Michiel'a Hazewinkel'a.

::SLA-RVIII-§10:: Redakcja przeczytała pojęcia oraz dowód twierdzenia, zna podstawy algebry homologicznej, topologiczne pojęcie homotopii, ale czym są kohomologie?

::SLA-RVIII-§11:: Po pobieżnym przemyśleniu wielomianów zredukowanych, odróżnieniu wielomianu zerowego od generujących zerowe funkcje, Redakcja przeczytała dowód

twierdzenia ::SLA-RVIII-§11-Tw.18:: o algebraicznej niezależności homomorfizmów grup w addytywną gupę ciała (Emil Artin). Po przeczytaniu pierwszych linijek dowodu ::SLA-RVIII-§11-Tw.19::, Redakcja nie mogła znaleźć błędu w pamięciowym, krótszym przecież rozumowaniu:

Jeśli s1,...,sn, parami różne, tworzą grupę automofizmów nieskończonego ciała K, to, zgodnie z ::SLA-RVIII-§1-Tw.2::, nad ciałem elementów stałych są grupą Galois. Gdyby dwa charaktery typu s1^(n1)*...*sn^(nn) były równe, wówczas po przeniesieniu na tą samą stronę, uporządkowaniu, dla każdego x<>0 w K, otrzymalibyśmy s1^(m1)(x)*...*sn^(mn)(x) = 1, czyli dla x z ciała elementów stałych: x^(m1+...+mn) = 1, a to ciało jest nieskończone, ponieważ, ponownie z ::SLA-RVIII-§1-Tw.2::, nieskończone ciało K jest nad nim skończone.

No tak: 1/s(x) = s(1/x) <> s^(-1)(x). I Redakcja przeczytała kolejny dowód Emil'a Artin'a.

::SLA-RVIII-§12:: Czy nie zrobić zestawienia punktów "specjalnych" rozszerzeń (twierdzenie o elemencie prymitywnym w ::SLA-RVII-§6::, twierdznie o bazie normalnej w ::SLA-RVIII-§12::)?

Zadania.

Jest to rozdział z największą ilością zadań w całej książce. Trudnych.

(w prawym, górnym rogu "Seiten", na trzecich stronach spisy treści):
https://gdz.sub.uni-goettingen.de/id/PPN243919689_0172
https://gdz.sub.uni-goettingen.de/id/PPN243919689_0173
https://gdz.sub.uni-goettingen.de/id/PPN243919689_0174

https://gdz.sub.uni-goettingen.de/

::FLRRiC:: Franciszek Leja "Rachunek różniczkowy i całkowy"

| | | | |
|---|---|---|---|
| Numer: | **MATINF 3/2023** | Stron: | 10 |
| Data wydania: | 30 października 2023 | Druk: | 3 darmowe egzemplarze |
| Adresy Redakcji: | czasopismo.matinf@gmail.com | Witryny informacyjne, regulaminy: | https://github.com/czasopismo-MATINF/czasopismo-MATINF |
| Czytelnik: | osoba samodzielnie ucząca się, student | | |
| Cel: | systematyczne kursy podstaw programowania, matematyki, algebry i analizy matematycznej, artykuły o takowej tematyce; recenzje książek do nauki, narzędzi do programowania, listy zasobów informatycznych, serwisów internetowych; po dwudziestu latach wybuchu programistycznego niepotrzebna już promocja programowania, ale powrót na jego podstawowy poziom i znalezienie miejsca dla jego podstaw przy potwornie zwiększającym się stopniu skomplikowania algorytmów oraz narzędzi programistycznych; systematyczny zbiór powszechnie znanych pomysłów i idei; | | |


## Wstęp do bieżącego numeru:

W tym numerze listopadowym kolejne sprawozdania z kursów programowania poprzez języki Python oraz Java, szkice z kursów analizy matematycznej oraz algebry. Również: lista najstarszych europejskich uniwersytetów, notatka o książkach do biologii, ciąg dalszy leksykonu.

## Spis aktualnie rozwijanej zawartości:

# Notatka 2. Wydziały, szkoły, instytuty lub fakultety matematyczne najstarszych uniwersytetów w Europie z różnych źródeł zebrane.

do XIII wieku (z pominięciem szkoły medycznej w Salerno):

Bologna
https://www.unibo.it/it
https://matematica.unibo.it/it/index.html

Paris
https://www.ac-paris.fr/
https://www.sorbonne.fr/

Oxford
https://www.ox.ac.uk/
https://www.mpls.ox.ac.uk/

Modena
https://www.unimore.it/
https://www.fim.unimore.it/site/home.html

XIII wiek:

z północy na południe i z powrotem:

Cambridge
https://www.cam.ac.uk/
https://www.maths.cam.ac.uk/

Valladolid
https://www.uva.es/export/sites/uva/
https://directorio.uva.es/arbol/2001
https://directorio.uva.es/arbol/2019
https://directorio.uva.es/arbol/2051

Salamanca
https://www.usal.es/
https://mat.usal.es/
https://diarium.usal.es/dptodmyce/
https://campus.usal.es/~matapli/

Sevilla
https://www.us.es/
https://matematicas.us.es/

Alcalá de Henares
https://www.uah.es/es/
https://www.uah.es/en/conoce-la-uah/campus-centros-y-departamentos/departamentos/Physics-and-Mathematics/

Madrid
https://www.ucm.es/
https://matematicas.ucm.es/
https://www.ucm.es/directorio?eid=3179
https://www.ucm.es/directorio?eid=3205

Murcia
https://www.um.es/
https://www.um.es/web/dp-matematicas/
https://www.um.es/web/didactica-matematicas/

Coimbra
https://www.uc.pt/
https://www.uc.pt/fctuc/dmat/

Lisboa
https://www.ulisboa.pt/
https://math.tecnico.ulisboa.pt/index.php

Lleida
https://www.udl.cat/ca/
https://dmat.udl.cat/ca/

Toulouse
https://www.univ-toulouse.fr/
https://www.math.univ-toulouse.fr/fr/

Montpellier
https://www.umontpellier.fr/
https://maths-fds.edu.umontpellier.fr/

Orléans
https://www.univ-orleans.fr/fr/

Angers
https://www.univ-angers.fr/fr/index.html
https://www.univ-angers.fr/en/education/schools-and-faculties/list/faculty-of-science.html

Italia, od południa:

Napoli
https://www.unina.it/home
http://www.matematica.unina.it/

Pontifical Academy of Sciences
https://www.pas.va/en.html
https://www.pas.va/en/taglist.disciplines.mathematics.html

Perugia
https://www.unipg.it/
https://www.dmi.unipg.it/

Siena
https://www.unisi.it/
https://www.diism.unisi.it/it/

Macerata
https://www.unimc.it/it

Padova
https://www.unipd.it/
https://www.math.unipd.it/

Parma
https://www.unipr.it/
https://smfi.unipr.it/it/

W zestawieniu Redakcja uwzględniła uniwersytety bez dłuższych przerw w działaniu. Uniwersytety Vercelli, Vicenza, Piacenza, Arezzo z oraz okolic "Doliny uniwersytetów nad Padem" (<>"Dolina zamków nad Loarą") zostały zamknięte, pierwszy uniwersytet w Palencia również, ale trwają przeniesione uniwersytety w Valladolid oraz Salamanca, podobnie uniwersytet w Northampton. Uniwersytet w Alcalá de Henares został przeniesiony do Madrytu. Uniwersytet Paryża jest podzielony na mniejsze. Redakcja nie znalazła stron wydziałów na tym oraz uniwersytecie w Orleanie, ani na uniwersytecie w Macerata.

https://pl.wikipedia.org/wiki/Vercelli
https://pl.wikipedia.org/wiki/Piacenza
https://pl.wikipedia.org/wiki/Vicenza
https://pl.wikipedia.org/wiki/Arezzo
https://pl.wikipedia.org/wiki/Northampton

Salerno
https://www.unisa.it/
https://www.dipmat.unisa.it/

Roma (już od 1303 roku)
https://www.uniroma1.it/
https://www.uniroma3.it/
https://www.mat.uniroma1.it/
https://www.mat.uniroma2.it/
https://matematicafisica.uniroma3.it/

## Notatka 2. Książka do biologii.

Redakcja nie będzie rozstrzygać pomiędzy grubymi książkami o tytułach "Biologia": czy lepsza jest Campbella, czy Villego. Prawdopodobnie, każda z tych książek zastępuje wszystkie podręczniki do biologii z dwunastu lat kształcenia. Mają z premedytacją więcej niż pięćdziesiąt dwa rozdziały, co oznacza, że czytając jeden rozdział tygodniowo, a chodząc na spacery, nie da się takiej książki przeczytać w jeden rok. By nie tracić wiosennego, letniego oraz jesiennych czasów, plan najlepiej rozłożyć na dwa lata: po połowie książki przez dwie kolejne zimy. Redakcji zdarzyło się również czytać książkę w języku angielskim "Life. The Science of Biology." dołączoną do kursów biologii na stronie MIT, ale odradza czytanie książek naukowych w obcych językach przez straty na czas tłumaczenia terminów naukowych. Oczywiście mikroskop.

Niemożliwe do zrozumienia są bazy danych genetycznych.

## Kurs programowania poprzez język Python.

Sprawozdanie 3.

Internet jest liczniej zapakowany stronami z materiałami do nauki języków niż dwadzieścia lat temu. Prawdopodobnie, Czytelnik już był przeczytał kilka rozdziałów z wybranej książki o języku Python oraz drugiej - o algorytmach, a Redakcja spróbuje opisać jego podstawowe elementy.

Większość do XX wieku skonstruowanych imperatywnych języków programowania zawiera instrukcje warunkowe oraz pętle. W Pythonie, przy if (jeżeli), match (dopasuj), for (od ... do ...), while (dopóki) nie używa się nawiasów, a przypisane bloki zaznacza wcięciami (kurs zaleca cztery spacje, ale n.p. środwisko Processing nie protestuje tabulatorom). W różnych językach instrukcje mogą się różnie nazywać. Instrukcje pomocnicze: break (przerwij), continue (kontynuuj), pass (nic); instrukcji jest więcej. By je poznać, dobrze jest przejrzeć listę słów kluczowych języka, znaczenie których poznaje się w trakcie dalszej nauki, ale ku zaskoczeniu Redakcji - instrukcji match na takiej liście dla Pythona nie ma. Słowa te świadczą nie tylko o proceduralności języka Python, ale też o obiektowości.

Chyba w każdym języku programowania występują typy oraz zmienne (pamięć !) (type, variable). W języku Python typ zapamiętanej wartości (value) nie jest oznaczony przy nazwie zmiennej, ale jest sprawdzany podczas działania programu. Typy zmiennych, którymi można się posługiwać, to najczęściej: liczby, napisy, tablice, klasy, wartości True/False. Z wbudowanymi kolekcjami: listami (tablice o zmiennej długości), tuple'ami (listy, w których nie można zmieniać elementów ani ich długości), słownikami, z użyciem języka Python można szybko napisać prosty, użyteczny program. Zdaniem Redakcji, korzystanie z napisów przypomina korzystanie z klas (typy definiowane przez Programistę); słowa "str" nie ma na liście słów kluczowych, mimo to napisy są dostępne bez dołączania modułów, zatem można je uznać za wbudowane w język.

Zamiast opisywać instrukcje Redakcja postanowiła rozpocząć programowanie skryptu do wyświetlania kwadratowych plansz do gry w środowisku Processing oraz klasy robota, który będzie się po tych planszach poruszał. Processing zawiera uproszczoną wersję języka Python (m.in. bez instrukcji match), a ściągawkę Czytelnik może zrobić z wykorzystaniem dokumentacji (reference), którą stopniowo należy przeczytać w całości. Redakcja próbuje trzymać się stylu kodowania zalecanego przez kurs. Processing jest jedynym znanym Redakcji środowiskiem, w którym przy pierwszym programowaniu można od razu wyświetlać obrazki (!).

Podczas pisania programu Redakcji sprawiało kłopoty, gdy nie działał, a brak było informacji zwrotnej o błędzie w dołączonych plikach - wtedy wyłączała fragmenty kodu komentarzem (#) i szukała błędu wiersz po wierszu. Program nie wyszedł zgrabnie; raczej przypomina t.zw. kod szkieletowy (skeleton code, dummy code) w Javie, być może z powodu zbyt małej ilości czasu poświęconej na myślenie o kolekcjach. Redakcja będzie skrypt uaktualniać, również ściągawkę języka Python, obie współbieżnie z czytaniem kursu. W środku metody draw jest miejsce na własny program Czytelnika do sterowania robotem. Czytelnik może poczytać o projektowaniu obiektowym.

Redakcja rozpoczęła pisanie skryptu testującego informacje z przygotowanej ściągawki oraz jej porządkowanie. Zamiast przeglądać skrypt Czytelnik powinien z możliwościami Pythona sukcesywnie zapoznawać się samemu, na odpowiedni dla siebie sposób systematyzując własną ściągawkę.

słowa kluczowe języka Python:
https://docs.python.org/2.5/ref/keywords.html
https://www.w3schools.com/python/python_ref_keywords.asp
styl kodowania:
https://peps.python.org/pep-0008/

https://docs.python.org/3/tutorial/controlflow.html
https://py.processing.org/reference/

# Kurs programowania poprzez język Java.

## Sprawozdanie 3.

Stron z materiałami do nauki języków jest o wiele więcej niż dwadzieścia lat temu. Prawdopodobnie, Czytelnik już był przeczytał kilka rozdziałów z wybranej książki o języku Java oraz drugiej - o algorytmach, a Redakcja spróbuje opisać jego podstawowe elementy.

Większość do XX wieku skonstruowanych imperatywnych języków programowania zawiera instrukcje warunkowe oraz pętle. W Javie, przy if (jeżeli), switch (dopasuj), for (od ... do ...), while (dopóki) używa się nawiasów okrągłych, a przypisane bloki zaznacza nawiasami wąsatymi. W różnych językach instrukcje mogą się różnie nazywać. Instrukcje pomocnicze: break (przerwij), continue (kontynuuj); instrukcji jest więcej. By je poznać, dobrze jest przejrzeć listę słów kluczowych języka, znaczenie których poznaje się w trakcie dalszej nauki. Razem z funkcjami słowa te świadczą o proceduralności oraz obiektowości języka Java.

Chyba w każdym języku programowania występują typy oraz zmienne (pamięć !) (type, variable). W języku Java typ zapamiętanej wartości (value) jest oznaczony przy nazwie zmiennej i, chyba z wyjątkiem rzutowań, jest sprawdzany podczas budowania wykonywalnego programu, co powoduje, że m.in. nie można zapamiętać liczby tam, gdzie obiecujemy napis. Typy zmiennych, którymi można się posługiwać, to najczęściej: liczby, napisy, tablice, klasy, wartości true/false. Z rozbudowaną biblioteką (frameworkiem) kolekcji i algorytmami operującymi na strumieniach język Java wydaje się być stworzonym dla zawodowych programistów. Firmy tworzą własne biblioteki o funkcjonalności dostosowanej do różnych problemów, jednak korzystanie z nich zwykle wymaga znajomości programowania generycznego (<>szablonów) i wykorzystania zaawansowanego środowiska programistycznego (IDE).

Podobnie do kursu programowania poprzez język Python w tym czasopiśmie, zamiast opisywać instrukcje, Redakcja postanowiła rozpocząć programowanie skryptu do wyświetlania kwadratowych plansz do gry w środowisku Processing oraz klasy robota, który będzie się po tych planszach poruszał. Processing zawiera uproszczoną wersję języka Java. Redakcja nie będzie robić ściągawek: część kodu będzie przygotowywać w IDE Eclipse, które, być może, dzięki statycznemu typowaniu języka Java, podaje listy dostępnych metod (Redakcja nazywa to programowaniem z użyciem kropki, dot programming), a dokumentację Processinga Czytelnik może właściwie przeczytać w całości. Processing jest jedynym znanym Redakcji środowiskiem, w którym przy pierwszym programowaniu można od razu wyświetlać obrazki (!). Redakcja spróbuje trzymać się znanego jej stylu kodowania - o profesjonalnych Czytelnik może poczytać.

Podczas pisania Redakcja miała problem: metody rysowania Processinga nie były widoczne poza głównym plikiem skryptu, więc umieściła cały kod w jednym pliku. Tablice Javy są mniej eleganckie niż listy Pythona; udało się skorzystać z instrukcji switch. Redakcja będzie skrypt uaktualniać. W środku metody draw jest miejsce na własny program Czytelnika do sterowania robotem. Czytelnik może poczytać o projektowaniu obiektowym.

słowa kluczowe języka Java:
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html

https://processing.org/reference/

## Leksykon 2. Paradygmaty.

dotyczące typów danych:

cechy typów:
statyczne oraz dynamiczne typowanie (rozpoznawanie typów zmiennych przed lub po uruchomieniu programu)
typowanie słabe oraz silne, typy węższe oraz szersze (automatyczne rzutowania)
typy proste, atomowe, złożone (primitive, atomic, composite)
typy oraz zmienne modyfikowalne oraz niemodyfikowalne (mutable, immutable)
typy wbudowane, typy użytkownika (user-defined)

typy:
logiczne, liczbowe, stało oraz zmienno-przecinkowe, całkowite (bool, boolean, numeric, decimal, fixed point, floating point, integer)
napisowy (string, text), wektorowy, macierzowy
wskaźnikowe, referencyjne, funkcyjne (pointer, reference, function)

budowanie innych typów:
typy podstawowe, typy złożone, szablony, typy generyczne (basic, fundamental, composite, template, generic)
kolekcje, sekwencje, zbiory (collections, sequences, sets)
unie, struktury, rekordy, obiekty (union, record, object)
wyliczeniowe (enumeration)
mixins

typ abstrakcyjny - każdy typ ze zdefiniowanym zbiorem operacji na wartościach typu, każdy typ
typ algebraiczny - zdefiniowany algebraicznie na podstawie typów bazowych (unie, rekordy, ...)
z ograniczeniami (quantified, refinement)

typy typów (meta)

Odkrywanie lub wymyślanie własnych paradygmatów pomaga zrozumieć zamierzone cechy dzieł.

Czy wymyślenie, czy odkrycie, czy też powstanie nowego paradygmatu są napędem technicznego postępu?

paradygmaty dotyczące wykorzystania produktu:
łatwy do przeniesienia, złożenia, zainstalowania, konfiguracji (configurable, scriptable)
łatwy do rozbudowania (extenstions, modules, plugins, scripts)
łatwy do wykorzystania, zrozumienia i nauczenia, zwięzły, treściwy, łatwy do przetłumaczenia
łatwy do zarządzania, utrzymania, naprawy
elegancki, szykowny, gustowny, wytworny, dystyngowany, wykwintny, ze smakiem, z fasonem, w dobrym tonie
praktyczny, poręczny, potrzebny, przydatny, ułatwiający

z przeglądania ustawień przeglądarek internetowych: configurability, understandability, learnability, compactibility, usability
safeability, trustability

# Kurs analizy matematycznej.

## Szkic 3. ::GMFI-RII::

Jako początkowy rozdział tej książki, ::GMFI-RII:: jest, w dalszym ciągu, powtórką polskiej szkoły średniej lat dziewięćdziesiątych. Być może, całą książkę czyta się szybko a rozumie łatwo. Nie zawiera aksjomatycznego, poukładanego, ujętego w system, krok po kroku wyprowadzenia ani definicji funkcji elementarnych, ani ich podstawowych wzorów, ale maluje ładne i czytelne, jak na druk sprzed rewolucji mikrokomputerowej, chyba ploterem rysowane wykresy. Czytelnik może wynotować wszystkie klasy funkcji elementarnych, nadając własne skróty niektórym niewymienionym cyklometrycznym. ::GMFI-RII-§2:: opisuje granice według ujęć Heine'go (Heinrich Eduard Heine) oraz Cauchy'ego (Augustin Louis Cauchy), a siłą książki objawia się w przykładowych wyprowadzeniach granic elementarnych potrzebnych w teorii ciągłości funkcji. Dla Czytelnika notującego: przynajmniej trzy wzory są używane w dowodach interesujących wzorów w dalszej treści książki, a tu wyprowadzenia są elementarne, zanim Czytelnik zacznie korzystać z metody różniczkowej de L'Hôpital'a (Guillaume François Antoine de L'Hôpital). ::GMFI-RII-§3:: opisuje geometryczne pojęcie nieskończenie małej. Zdaniem Redakcji, jeśli Czytelnik zastanowi się i opracuje metodę ich dzielenia, wówczas czytanie o formach różniczkowych w przyszłości będzie znacznie łatwiejsze. Zrozumienie porównywania rzędów nieskończenie małych pomaga w rozumieniu operowania wzorami na wykładach z fizyki.

::GMFI-RII-§4:: wprowadza dwa typy punktów nieciągłości funkcji, ale ich rozróżnienie, o ile Redakcja dobrze pamięta, przydaje się przy porównaniu całki na przestrzeniach R^n w ujęciach Riemann'a (Georg Friedrich Bernhard Riemann) i Lebesgue'a (Henri Léon Lebesgue). W tym rodziale pojawiają się równania funkcyjne, które są jedną z trzech klas równań znanych Redakcji próbowanych być objętymi dużymi teoriami matematycznymi: algebraiczne, różniczkowe, funkcyjne. ::GMFI-RII-§5:: to, przede wszystkim, zestaw pokrewnych twierdzeń o przyjmowaniu wartości średnich i kresowych, ale również pojęcie zbieżności jednostajnej wprowadzające przestrzenie funkcyjne, którymi, z pozoru, zajmuje się analiza funkcjonalna. Dowody są łatwe do zrozumienia i narysowania (!), co odróżnia te od formalnego wykładu teorii przestrzeni Banacha (Stefan Banach). Podobnie, Czytelnik znający podstawy topologii, znajdzie lemat Borela (Émile Borel) prostym. Redakcja doradza robić notatki w zeszycie formatu B5 w kratkę, co najmniej stu kartkowym, z twardą okładką, chociażby dlatego, że wertowanie zeszytu jest o wiele łatwiejsze niż dla sprawdzenia przeszukiwanie przeplatanych z wyprowadzeniami wzorów w książce. Redakcja poleca również korzystanie z dużej, białej, suchościeralnej tablicy na alkoholowe flamastry.

# Kurs algebry.

## Szkic 3. ::SLA-RVIII:: Teoria Galois.

::SLA-RVIII-§1:: Redakcja nie będzie komentować.

Przykłady ::SLA-RVIII-§2:: są testem jego dokładnego oraz szczegółowego zrozumienia. Pobieżne czytanie prowadzi do błędnego, obecnie zdaniem Redakcji, przekonania, że wnioski z twierdzenia Galois wyprowadza się łatwo, chociaż właśnie wtedy widać ogólny krajobraz możliwości. Błądzenie po teorii Galois można porównać do górskiej wycieczki. Stopień rozszerzenia ciała k(t1,...,tn) do elementów stałych grupy symetrycznej Sn w przykładzie czwartym wynika z rzędu tej grupy n!, wszystkie elementy t1,...,tn są algebraiczne nad ciałem k(s1,...,sn). Przykład piąty nawiązuje do treści ::SLA-RIX::, ale Redakcja nie zaleca przeskoczenia i czytania tego rozdziału w tym momencie, ponieważ zawiera ogólny opis porządków, elementów nieskończenie małych oraz dużych, które przy liczbach rzeczywistych i zespolonych chyba nie występują (?). Do uogólnienia tego przykładu można wtedy wrócić.

::SLA-RVIII-§3:: jest analizą równania X^n - 1 = 0 równoważnego w ciałach X^(n+1) = X. W ::SLA-RVIII-§5:: wprowadzone są pojęcia śladu oraz normy wykorzystane w ::SLA-RVIII-§6:: oraz ::SLA-RVIII-§9::, które są analizą równania X^n - a = 0. W ::SLA-RVIII-§3::, w dowodzie twierdzenia szóstego kongruencje wielomianów wynikają z dowolnie dużej liczby pierwszej p (>n) tak, by redukcja h-fg była trywialna oraz miał on więcej pierwiastków niż wynosi jego stopień (czy waluacja wielomianu w jednym punkcie decyduje o pierwiastkach w innych ?). Rozumowania przejścia od Q do innych ciał, przykłady oraz twierdzenie, które następują, wykorzystują Lemat Gaussa (Johann Carl Friedrich Gauß) z ::SLA-RV-§6:: oraz podstawowe fakty i wzory teorii liczb, które można znaleźć n.p. w ::WSTL:: lub z dowodami opartymi o odwracalność w splotach w ::WNTL:: : małe twierdzenie Fermat'a (Pierre de Fermat), wzory funkcji Euler'a (Leonhard Euler), funkcji Möbius'a (August Ferdinand Möbius), symbolu Legendre'a (Adrien-Marie Legendre). Do przykładu trzeciego Redakcja zgłasza wątpliwość: f2 = X + 1. Po nich wymieniony jest ładny wzór sumacyjny, który dla liczb zespolonych można uzasadnić geometrycznie w oparciu o symetrie wielokątów gwiaździstych, oraz twierdzenie o pewnej relacji rozszerzeń kwadratowych oraz cyklotomicznych Q.

Czy twierdzenie Artin'a (Emil Artin) z ::SLA-RVIII-§4:: ma coś wspólnego z teorią waluacji - "homomorfizmów" ciał w grupy uporządkowane odpowiadającym częściowym homomorfizmom ciał z nieskończonością (t.zw. place'om, patrz ::SLItAG-ChI::)?

Czy prócz normy (mnożenie, jedynka) oraz śladu (dodawanie, zero) pozostałe funkcje symetryczne pierwiastków wielomianu mają odpowiadające działania oraz elementy neutralne? Jak zdefiniować funkcje - by e^Tr = N(e)? Obliczenia pierwiastków na wektorach z iloczynem skalarnym. Twierdzenie o nietrywialności śladu porównać z motywem dualności w ::SLA-RI-§11:: i ::SLA-RIII-§6:: . W dowodzie wniosku drugiego Redakcja musiała skorzystać z transpozycji macierzy oraz wyznacznika; dlaczego założenia wymagają rozdzielczości? W dowodzie stwierdzenia pierwszego wybór pierwiastka wielomianu minimalnego następuje na końcu.

::SLA-RVIII-§6::, ::SLA-RVIII-§7:: są proste oraz przyjemne - w odróżnieniu od ::SLA-RVIII-§9::, jeśli bez rozpisania a zdaniem Redakcji skomplikowane, zawierającego indukcyjny dowód pewnego kryterium nierozkładalności X^n-a oraz jedno z licznych twierdzeń Artina, którego książek Czytelnik sam poszuka w internecie, oraz Schreiera (Józef Schreier). Ponownie zdaniem Redakcji, łatwiej udowodnić tożsamość klas rozszerzeń rozwiązalnych oraz pierwiastnikowych korzystając z ::SLA-RI-Ćw13::, a o pierwiastnikowych łatwo już - że są klasą wyróżnioną. Należy pamiętać o rozszerzaniu do normalnego oraz korzystać z pięter konstruowanych na podstawie ::SLA-RI-§4-Stw4:: .

::WSTL:: Wacław Franciszek Sierpiński "Teoria liczb"
::WNTL:: Władysław Narkiewicz "Teoria liczb"
::WNEATL:: Władysław Narkiewicz "Elementy algebraicznej teorii liczb"
(łatwiejsza niż wzory w popularno-naukowych książkach prof. W.Sierpińskiego, jedyne proste, w języku polskim, znane Redakcji omówienie rozszerzeń liczb całkowitych ze wstępem do niesamowitych pierścieni Dedekinda (Julius Wilhelm Richard Dedekind) definicji oraz teorii, zrozumiała dla uczniów szkół średnich)

::SLItAG:: Serge Lang "Introduction to Algebraic Geometry"

| | | | |
|---|---|---|---|
| Numer: | **MATINF 2/2023** | Stron: | 11 |
| Data wydania: | 25 września 2023 | Druk: | 3 darmowe egzemplarze |
| Adresy Redakcji: | czasopismo.matinf@gmail.com | Witryny informacyjne, regulaminy: | https://github.com/czasopismo-MATINF/czasopismo-MATINF |
| Czytelnik: | osoba samodzielnie ucząca się, student | | |
| Cel: | systematyczne kursy podstaw programowania, matematyki, algebry i analizy matematycznej, artykuły o takowej tematyce; recenzje książek do nauki, narzędzi do programowania, listy zasobów informatycznych, serwisów internetowych; po dwudziestu latach wybuchu programistycznego niepotrzebna już promocja programowania, ale powrót na jego podstawowy poziom i znalezienie miejsca dla jego podstaw przy potwornie zwiększającym się stopniu skomplikowania algorytmów oraz narzędzi programistycznych; systematyczny zbiór powszechnie znanych pomysłów i idei; | | |


## Wstęp do bieżącego numeru:

W tym numerze październikowym kolejne sprawozdania z kursów programowania poprzez języki Python oraz Java, szkice z kursów analizy matematycznej oraz algebry. O możliwych modyfikacjach języków programowania, trójwymiarowych wrażeniach rzeczywistości generowanych komputerowo oraz agencjach kosmicznych, część pierwsza leksykonu.

## Spis aktualnie rozwijanej zawartości:

# Szkic 1. O możliwych typach modyfikacji języków i typach języków programowania.

Każdy programista korzysta ze zintegrowanego środowiska programistycznego (?) (!) (IDE, Integrated Development Environment). Wszystkie programy pisane są w językach programowania, do których słów kluczowych a i nazw tworzonych komponentów (procedur, funkcji, modułów) używane są słowa języka angielskiego. Niewielu zastanawia się nad tym, a o usprawnienia ze szczyptą kreatywności - łatwo - kolor i wielkość czcionki, podkreślenie, przekreślenie mogłyby rozbudować semantykę programu. Dla przykładu:

`for(i = 0; i < 10; i++) {...}` - wykonaj sekwencyjnie

`FOR(i = 0; i < 10; i++) {...}` - wykonaj współbieżnie

Obiecujące wydaje się wykorzystanie innych zestawów znaków. Nie tylko "alfabety" z dużą ich ilością (n.p. chiński lub japoński), w których pojedynczym znakom przypisać można różne algorytmy rozwijając semantykę i skracając długość programowania funkcyjnego:

`[...]` 国 立 研 - wykonaj kolejno operacje 国 立 研 na strumieniu danych `[...]` (skrócony zapis wykonania n.p. `[...].sort().distinct().first()` z języka Java)

`int` 究 開 発`;` - większa ilość znaków to większa ilość typów oraz zmiennych o krótszych i piktograficznych nazwach

Piktograficzny "alfabet" programowania, w którym położenie elementów znaków względem siebie niesie znaczenie. Dla przykładu:

法 - trzy elementy po lewej oznaczają pozycje parametrów procedury oznaczonej elementem po prawej

人 航 - odbij obiekt graficzny 航 symetrycznie, postaw go po lewej obiektu odbitego

宇 空 - stwórz ciąg pomniejszanych obiektów graficznych 空 - mniejsze nad większymi - aż do zniknięcia

宙 10 20 研 - stwórz tablicę o wymiarach 10x20 z obiektów 研

Widać, że tak programować łatwiej rysując na tablecie graficznym lub kartce niż klawiaturze. Ze względu na łatwość tworzenia graficznych interfejsów użytkownika oraz dokonywania zmian w programowaniu ciężko uwierzyć, czy duże firmy nie mają takich narzędzi?

Programy można również wytańczyć w języku migowym. Programista, niekoniecznie przed monitorem, sterując IDE przy pomocy głosu i gestów mógłby programować!

`@: nowa procedura "sortowanie"`

`@: weź ([...]) @: rozmiaru "sto"`

`@` - na głos, `(...)` - gest rękami

Powstaje potrzeba tłumaczenia między semantycznymi językami znakowymi maszyn (wirtualnych lub robotów) a migowymi oraz bliskimi naturalnym.

(!) Czy zwierzęta będą mogły programować? Czy da się odczytać programy rządzące ruchem pszczół? (!)

Przykłady Czytelnik może podać sam:

[* 0 - 100 : "a-r" *] - stworz tablicę indeksowaną od 0 do 100 i wypełnij ją losowymi literami z zakresu od 'a' do 'r'

究開 発構 機 - przejdź robotem 機 do ściany, wejdź na piętro, potem nadaj sygnał 構

(...)

Zdaniem Redakcji, wśród istniejących alfabetów na uwagę zasługują:

https://pl.wikipedia.org/wiki/Runy
https://pl.wikipedia.org/wiki/Alfabet_ormia%C5%84ski
https://pl.wikipedia.org/wiki/Hangul
https://pl.wikipedia.org/wiki/Pismo_dewanagari

# Kurs programowania poprzez język Python.

## Sprawozdanie 2.

Pomimo eleganckiego spisu treści, Redakcja zrezygnowała z czytania strony learnpython.org, ponieważ wydaje się prezentować porównanie podstaw różnych języków, a nie systematycznym kursem. Natomiast zbierając i porządkując linki do materiałów na stronie docs.python.org w trakcie czytania głównego tekstu można tworzyć poręczną bazę wiedzy oraz spojrzeć w dal na ścieżkę nauki. Dobrym pomysłem jest notowanie własnej ściągawki (reference sheet, cheat sheet).

Na znane Redakcji języki programowania składają się instrukcje (instructions), typy (types), operatory (operators, operations), funkcje (functions, procedures). Ten schemat powtarza się chyba dla każdego proceduralnego i obiektowego języka programowania, jakim jest Python. W takich można tworzyć własne funkcje, typy - klasy (classes), w niektórych własne operacje - przeciążanie operatorów (operator overloading), ale instrukcje zwykle są niezmienne. Podział jest nieco płynny, ponieważ operatory są - funkcjami o symbolicznych nazwach a instrukcje - o nazwach wbudowanych w język. Funkcja po prostu "coś robi". Pass.

W punktach 3 oraz 4 czytanego kursu opisano cztery powszechne instrukcje if, for, while, match (w innych językach - switch lub case). Instrukcje break, continue, pass są pomocnicze. Z tych punktów oraz dodatkowych linków do materiałów Redakcja zebrała:

typy liczbowe (number) : całkowity - int, (zmienno)przecinkowy - float, liczb zespolonych - complex

typ "napisowy" (string) : str

typy kolekcji (collections) : lista - list; tuple; słownikowy (dictionary) - dict; zbiorowe - set, niemodyfikowalny - frozenset; wyliczeniowe - range, slice

stałe (constants) : True, False, None

Redakcja zrobiła tu własny podział, niepełny i nieco niezgodny z proponowanym w kursie (numerics, sequences, mappings, classes, instances and exceptions).

Z wykorzystaniem instrukcji do wczytywania z klawiatury i wypisywania na ekranie informacji od i dla siedzącego przed komputerem jest wystarczającym zestawem do pisania prostych programów.

Redakcja przygotowała i zaraz wydrukuje jednostronnie pierwszą wersję ściągawki. Zamiast pobrania z internetu, przygotowanie pomocy samemu jest męczące, ale zakończona jest również planem nauki. Niestety, prawdopodobnie, nie tylko zdaniem Redakcji, przygotowanie listy podstaw języka samemu na podstawie dokumentacji lub treści internetowego kursu wymaga dobrej znajomości nie zaawansowanych, ale teoretycznych podstaw programowania. Redakcji zajęło to dwa dni. Napisanie rano (!) programu o odgadywaniu liczby - około półtorej godziny. Znajomość operacji na oraz abstrakcyjnych struktur danych tablic (array), list (list), stosów (stack), drzew (tree), pojęcia haszowania (hash), pojęcia iteratora (iterator), złożoności czasowych i pamięciowych, operatorów oraz ich priorytetów, teorii odwrotnej notacji polskiej (ONP) pozwala na szybkie rozeznanie się w SPL oraz podstawowych bibliotekach innych języków programowania, by wiedzieć której metody użyć. Zatem Czytelnikowi potrzebna jest druga papierowa książka o algorytmach, którą będzie czytać równolegle z książką o języku Python. Znajomość typowych instrukcji, pojęcia zmiennej, typów zmiennych - na szybkie płynne korzystanie ze ściągawki. Redakcja zalecałaby uczenie się tych tematów przy dużej białej suchościeralnej tablicy na alkoholowe flamastry.

Nie ma tu jednak operacji grafowych. Są bazy danych, ale redakcja nie zna języka programowania z wbudowanymi w podstawy operacjami przetwarzania grafów. (!)

Programowanie codziennych działań domowych oraz osiedlowych robotów - zamiast kalendarycznych formularzach na domowym panelu kontroli wyposażonym w język Python oraz domowe API - nie wymagałoby znajomości teorii. (!)

Ten akapit ustala plan pojęć - krok siedmiomilowy - w nauce programowania z wykorzystaniem języka Python:

list (list comprehension), tuple, for, while | tablice, listy, iteratory

str | napisy, algorytmy przetwarzania tekstu, kodowanie

if, match (wildcard, guard), boolean

literal, constant, variable, type | stałe, zmienne, typy, typy typów, rzutowania wartości zmiennych

expression, operator, priority, short circuit, function, procedure | wyrażenia, operatory, funkcje i procedury

dict, unpacking | słowniki, drzewa, haszowanie

class, object, interface, equality | klasy, obiekty, interfejsy, porównywanie

https://docs.python.org/3/library/stdtypes.html
https://docs.python.org/3/library/functions.html
https://docs.python.org/3/library/constants.html

# Notatka 1. O agencjach kosmicznych na Ziemi.

Lista agencji kosmicznych, które zorganizowały loty załogowe lub są do tego gotowe (według https://pl.wikipedia.org/wiki/Lista_agencji_kosmicznych):

Loty załogowe:

Государственная корпорация по космической деятельности «Роскосмос» (Roskosmos, RFSA, https://www.roscosmos.ru/)
国家航天局 (Chińska Narodowa Agencja Kosmiczna, CNSA, https://www.cnsa.gov.cn/)
National Aeronautics and Space Administration (Narodowa Agencja Aeronautyki i Przestrzeni Kosmicznej, NASA, https://www.nasa.gov/)

Gotowość do lotu załogowego:

European Space Agency (Europejska Agencja Kosmiczna, ESA, https://www.esa.int/)
Державне космічне агентство України (Państwowa Agencja Kosmiczna Ukrainy, NSAU, https://www.nkau.gov.ua/ua/)
סוכנות החלל הישראלית (Izraelska Agencja Lotów Kosmicznych, ISA, https://www.space.gov.il/)
भारतीय अंतरिक्ष अनुसंधान संगठन (Indyjska Organizacja Badań Kosmicznych, ISRO, https://www.isro.gov.in/)
国立研究開発法人宇宙航空研究開発機構 (Japońska Agencja Eksploracji Aerokosmicznej, JAXA, https://global.jaxa.jp/)

# Kurs programowania poprzez język Java.

## Sprawozdanie 2.

Zatem jest rok 2023. Programiści programują przy komputerach z ekranami oraz laptopach (przenośnych komputerach z ekranami). Środowiska programistyczne są uruchamiane na małym komputerze lub w rozproszonych centrach obliczeniowych (chmurach - clouds), a małe komputery łączą się z tymi środowiskami poprzez "sieć internetową" przesyłającą dane poprzez przestrzeń w oparciu o zasady elektromagnetyzmu. Popularnych języków programowania dostosowanych do innych potrzeb wyrazu treści jest chyba kilkadziesiąt. Jeszcze więcej jest narzędzi programistycznych, stworzonych bibliotek, API-ów. Duża część była pisana ręcznie.

Zapomniany do poprzedniego artykułu link do Processing'a:

https://processing.org/

Przejrzeć przykłady dołączone do środowiska Processing w menu "File", ale bez matematyki zrozumieć, dlaczego tak działają, jest chyba niemożliwym. Podstawowy program w środowisku Processing zawiera metodę początkową (setup), w środku której ustalamy rozmiar okna (size(poziomy, pionowy), od kiedy Redakcja pamięta współrzędne w programach liczone są od lewego górnego kąta płótna) oraz metodę obliczajacą i wyświetlającą kolejne klatki filmu (draw). Należy wyświetlić ściągawkę (reference), której Redakcja nie znalazła przetłumaczonej na język polski w wersji do wydrukowania.

https://processing.org/reference/

Redakcja nie przeglądała, ale jest prawie zupełnie pewna, że materiały na podstronach przykładów, kursów oraz książek są baardzo wysokiej jakości.

https://processing.org/examples/
https://processing.org/tutorials/
https://processing.org/books/

Tym razem Redakcja napisała skrypt wyświetaljący kółka w losowych miejscach na płótnie (canvas).

Redakcja załącza kilka linków do stron z zaawansowanymi materiałami dotyczącymi języka Java z internetowej historii Redakcji:

https://docs.oracle.com/javase/8/ (materiały kursu nie tylko dla wersji 8 Javy na stronie Oracle Corporation)
https://javaconceptoftheday.com/ - zawiera tablice (reference sheets, cheat sheets), ale chyba lepiej przygotować samemu
http://fxexperience.com/ (strona z blogiem o bibliotece JavaFX do tworzenia GUI, ale bez kursu)
z tych stron nie udało się Redakcji nauczyć frameworka Spring:
tutorials/https://www.baeldung.com/
https://howtodoinjava.com/
https://javabeat.net/
https://www.dineshonjava.com/

# Notatka 1. ER, AR, MR, VR.

Redakcja nie wie na czym docelowo polegają projekty firm tworzenia rzeczywistości wirtualnej, ale wyobraża sobie następująco:

rozpoczynając podróż na rowerze zakładamy na głowę przezroczysty hełm lub okulary, przewiewne po bokach
dotknięcie czytnika odcisków palców łączy okulary z chmurą danych
hełm łączy się z komputerem pokładowym roweru, włącza światła, ustawia lusterko, raportuje o stanie roweru oraz bagażu
na świat rzeczywisty nakładaja się wyświetlane informacje (rozszerzona rzeczywistość, augmented reality, extended reality, mixed reality): metadane z chmury, monochromatyczne lub kolorowe
dojeżdżając do miasta symbole pokazują kierunki muzeów, parków, zadaszonych miejsc wypoczynkowych, parkingów, toalet, ...

Redakcja nie wie, czy istnieja giętkie "smartfony", przezroczyste ekrany z "genetycznie" modyfikowanego plastyku (technologia "quantum dots"?) podobne do butelek na wodę mineralną lub ramki do naklejenia rozpoznające geometrię powierzchni; nie zna również dokładnych zasad działania dostępnych okularów VR rzeczywistości wirtualnej (virtual reality), które nie są przeroczyste.

W 2023 roku okularów oraz oprogramowania do VR oraz AR jest chyba bez liku.

Takie projekty wydaje się mieć dużo firm, w tym za wikipedią: "HTC Corporation", "Magic Leap, Inc.", "Meta Platforms, Inc.", "Microsoft Corporation", "PICO", "Sony Interactive Entertainment", "Valve Corporation", "Varjo Technologies Oy", "zSpace". Listę Redakcja znalazła na stronie standardu:

https://en.wikipedia.org/wiki/OpenXR#Implementations

https://www.vive.com/eu/
https://www.magicleap.com/
https://about.meta.com/realitylabs/
https://www.microsoft.com/en-us/hololens/
https://www.picoxr.com/
https://www.playstation.com/pl-pl/ps-vr/
https://www.valvesoftware.com/pl/
https://varjo.com/
https://zspace.com/

Wykorzystywane są różne sposoby: rozpoznawanie położenia ciała przy pomocy zestawu kamer, obliczanie położenia kontrolerów zestawu przy pomocy czujników, kamery na podczerwień śledzące kierunek spojrzenia oczu, mikroprojektory montowane w okularach (!). Oprócz ofert zastosowań dla dużych firm reklamowane są aplikacje pomagające w prowadzeniu szkoleń oraz gry. Na stronie firmy "Varjo Technologies Oy" znajduje się obszerny słownik:

https://varjo.com/learning-hub/

Czy ER oraz MR będą takie, jak je zdefiniowano? (!)

Firma "zSpace" opracowała wyświetlanie nie wymagające okularów! Czy ekran rozpoznaje strumień elektromagnetyczny drążony z niego przez widza? (!)

Czy standardy API grafiki 3d powinny zawierać funkcje zwiększające szybkość lub jakość wyświetlania obiektów i scen? (seperation of concerns) (!)

Proporcje ceny do możliwości najlepiej sprawdzić samemu kilkoma wizytami w nie tak licznych studiach VR.

# Leksykon 1. Paradygmaty.

Jedno ze znaczeń słowa 'paradygmat' według internetowego słownika języka polskiego PWN: «przyjęty sposób widzenia rzeczywistości w danej dziedzinie, doktrynie itp.». Zdaniem Redakcji, firmy informatyczne, twórcy oprogramowania oraz języków programowania, reklamodawcy wydają się używać tego pojęcia narzucając błędnie podwójnie zarysowane, zamaskowane znaczenie: osiągnięta zamiast «zamierzona przez twórców cecha, właściwość, jakość, opinia», łatwość w ich uzyskaniu. Tworzenie własnego słownika paradygmatów produktów informatycznych znacznie poszerza kręgi patrzenia i widzenia, poziom zrozumienia technik, typów procesów, etapów, faz, cykli (?) wytwarzania oprogramowania, jakość kodu, działania i pracę w firmach nie tylko informatycznych.

imperative

structured, procedural

encapsulation:
object oriented, functional

flow:
event-driven (n.p. dla GUI), (message-driven)

concurrent, distributed

declarative:
constraints, logic

Każde znane Redakcji oprogramowanie zawiera paradygmat "imperative": działa w procesorze w oparciu o języki typu "assembler". Pozostałe paradygmaty są organizowaniem wyższych pięter. Programowanie strukturalne to wykorzystanie instrukcji "goto", proceduralne pozwala na porządkowanie kodu w metodach; różnica pomiędzy paradygmatami obiektowym a funkcjonalnym polega na zachowaniu przez obiekty stanu pomiędzy działaniem metod: zespół obiektowych robotów z pamięcią oraz zespół funkcjonalnych robotów bez pamięci. Wszystkie te paradygmaty można wyobrazić w sposób współbieżny oraz rozproszony (czas i miejsce). Sterowanie zdarzeniami (komunikatami) są sposobami przekazywania sterowania pomiędzy obiektami.

A może bardziej dopasowane nazwy: oznajmujące, rozkazujące (constraints) oraz pytające (logic) ? (!)

# Kurs analizy matematycznej.

## Szkic 2. ::GMFI-RI::

Jeśli Czytelnik lub Czytelnicy zdecydowali kupić książkę, co najmniej stukartkowy zeszyt formatu B5 w kratkę, przynajmniej dziesięć wysokiej jakości ołówków, gumkę oraz temperówkę, oraz ... znaleźli dużą białą tablicę suchościeralną na alkoholowe flamastry, to prawdopodobnie powtórzyli już liczby wymierne i rzeczywiste z wstępu. ::GMFI-RI:: przechodzi do rzeczy. Siłą książki nie jest teoria, ale obliczenia zawarte w przykładach. Nie zawiera ani aksjomatycznego wyprowadzenia zależności pomiędzy geometrią euklidesową (Ευκλείδης) a przestrzenią R^n, ani ścisłego wyprowadzenia pojęć powierzchni oraz objętości. To natychmiast zniechęca teoretyków. Przykłady rozrzucone po paragrafach w nich punktach należy grupować i notować bez dowodów. W trakcie czytania do każdej grupy notatek następne notatki z rozproszonych miejsc w trzech tomach. Zapisując n.t.p. granice należy zostawić kilka podwójnych kartek pustych na nowe. Zostawiając cztery lub pięć dla każdej grupy i tak któraś się nie zmieści, a zeszyt będzie zawierał przeplatane sekcje z informacjami tego samego typu.

Tak można zapisać cały zeszyt! Pomimo nietrwałości Redakcja preferuje pisanie ołówkiem umożliwiającym korektę oraz wielokrotne przeorganizowanie notatek. Po zatwierdzeniu można użyć kolorowych długopisów żelowych do wyodrębniania sekcji oraz zaznaczeń.

Proponowane przez Redakcję sekcje dla ::GMFI:: i ::GMFII:: do zaplanowania zeszytu: nierówności, granice, szeregi (kryteria zbieżności), trygonometria, przestrzenie funkcyjne, wzory całkowe, krzywe, różniczkowanie jednej zmiennej, różniczkowanie wielu zmiennych, rozwinięcia funkcji w szeregi. Książka zawiera interesujące wzory konkretnych funkcji, w tym t.zw. specjalnych, które Redakcja grupowała po stronie dla funkcji, co nie okazało się wystarczające dla późniejszych notatek. Redakcja zaleca notowanie również podstawowych wzorów na pochodne i całki do ich późniejszego porównania z zaawansowanymi.

Oprócz wyprowadzenia podstawowych granic ciągów potrzebnych do dowodów ciągłości oraz różniczkowalności funkcji elementarnych w ::GMFI-RII::, definicji oraz rozwinięcia w szereg liczby e, rozumowań o średnich Gaussa (Johann Carl Friedrich Gauß) ::GMFI-RI:: przedstawia twierdzenie Stolza (Otto Stolz) o możliwości zamiany obliczenia granicy ilorazu ciągów na granicę (!) ilorazów różnicowych (!) o ile pierwsza istnieje. Wnioskiem jest twierdzenie Cauchy'ego o podobnej zamianie na granicę średnich arytmetycznych.

Podobnie jak znajomość podstawowych faktów teorii przestrzeni Banacha oraz ich odwzorowań liniowych pomoże przy nauce różniczkowania funkcji wielu zmiennych, tak znajomość przestrzeni metrycznych, równoważności metryk i topologicznych pojęć zwartości oraz zupełności czynią ::GMFI-RI-§4:: oraz twierdzenie Bolzano-Weierstrassa (Bernard Bolzano, Karl Theodor Wilhelm Weierstraß) łatwym do zrozumienia od razu dla przestrzeni R^n. Przy granicach liminf oraz limsup warto zauważyć, że poniżej lub powyżej może występować nieskończenie wiele elementów. Po uwzględnieniu -inf oraz +inf stają się najmniejszym oraz największym punktem skupienia podciągów.

# Kurs algebry.

## Szkic 2. ::SLA-RVII:: ciała skończone.

Mnóstwo jest nie tylko nowych książek do algebry; w wielu językach. Być może niektóre z nich dopasują się do psychologicznych podstaw umysłu i staną się popularne. Redakcja nie czytała ich wiele, ale ze względu na ilość podręczników opublikowanych przez prof. S.Lang'a, rolę grupy "Nicolas Bourbaki" w porządkowaniu (!) podstaw matematyki XX wieku (!), Redakcja poleca tę książkę.

Tutaj Redakcja chciała napisać o równaniach X^n - a = 0, ale zauważyła, że są opisane dopiero w ::SLA-RVIII-§3:: oraz ::SLA-RVIII-§9:: .

Czytelnik miał wybór wielokrotnych możliwości: czytać powoli oraz robić dokładne notatki, rozumieć natychmiast i teraz jest już w następnych rozdziałach, pomijać dokładne czytanie dowodów, rozmawiać z innymi Czytelnikami, sprowadzać każde rozumowanie do wzoru wielomianowego. Redakcja zakłada, że po miesiącu Czytelnik rozumie pojęcie ciała rozkładu wielomianu (rozszerzenia normalnego).

Przy zastanawianiu się nad ciałami skończonymi - rozszerzeniami ciał Zp - najważniejszy jest wzór Newtona (Sir Isaac Newton) na (x+y)^n. Jego szkolny dowód wykorzystuje kombinatorykę, a współczynniki są liczbami całkowitymi powstającymi z sumowania podobnych składników, a stąd wynika poprawność w pierścieniach przemiennych. Współczynniki dwumienne obliczamy nad Z, potem redukujemy modulo p (wyobrazić mnożenie oraz dodawanie). Gdy p jest liczbą pierwszą, wtedy dzieli współczynniki dwumienne z wyjątkiem "pierwszego" i "ostatniego":

(x+y)^p = x^p + y^p oraz podobnie (x+y)^(p^n) = x^(p^n) + y^(p^n) (!)

Dwa zera x^q - x = 0, y^q - y = 0 mają ciekawą własność: (xy)^q = (x^q)(y^q) = xy => (xy)^q - xy = 0 : ich iloczyn też jest pierwiastkiem tego równania. Tworzą skończoną grupę. (!)

Ale również (x+y)^q = x^q + y^q = x + y. Tworzą q-elementowe ciało przemienne. (!)

Jednznaczność w ::SLA-RVII-§3-str.171-Tw.3:: .

X^p jest automorfizmem ciał skończonych. (!)

Czytelnik powinien wyobrazić (narysować diagram) piętra ciał odwzorowujące podzielność wykładników p : x^(p^n) = x^(p^(m+n-m)) = (x^(p^m))^(p^(n-m)) = x^(p^(n-m)) = ... = x^(p^0) = x (dla m | n !) - domknięcie algebraiczne ciała Zp. Ponieważ ciało jest skończone, to grupa automorfizmów jest podgrupą grupy permutacji, zatem dla rzędu jest (x^p)^n = x. Wszystkie elementy ciała spełniają równanie X^(p^n) = X. Rozdzielczość wynika z rozdziału ::SLA-RV-§10:: o rugowniku. Przemyśleć dla rozszerzeń Zp, potem Zq. Wszystkie piętra ciał skończonych są rozszerzeniami Galois (::SLA-RVIII-§1::).

Poprzez redukcję ilości działań w badanej strukturze teoria Galois opisuje relacje pomiędzy rozszerzeniem ciał (dodawanie i mnożenie) a grupą automorfizmów ciała większego (składanie), które nie poruszają elementów ciała mniejszego. Czy to jest uproszczenie? Grupa wydaje się prostszą strukturą, ponieważ ma tylko jedno działanie, ale może być nieprzemienna, natomiast dodanie zależności w definicji struktury zmniejsza ilość jej rzeczywistych modeli (Q oraz Zp). Czytając ::SLA:: Redakcja nie zajmowała się nieskończonymi rozszerzeniami Galois.

| | | | |
|---|---|---|---|
| Numer: | **MATINF 1/2023** | Stron: | 7 |
| Data wydania: | 28 sierpnia 2023 | Druk: | 3 |
| Adresy Redakcji: | czasopismo.matinf@gmail.com | Witryny informacyjne: | https://github.com/czasopismo-MATINF/czasopismo-MATINF |
| Czytelnik: | osoba samodzielnie ucząca się, student | | |
| Cel: | systematyczne kursy podstaw programowania, matematyki, algebry i analizy matematycznej, artykuły o takowej tematyce; recenzje książek do nauki, narzędzi do programowania, listy zasobów informatycznych, serwisów internetowych; po dwudziestu latach wybuchu programistycznego niepotrzebna już promocja programowania, ale powrót na jego podstawowy poziom i znalezienie miejsca dla jego podstaw przy potwornie zwiększającym się stopniu skomplikowania algorytmów oraz narzędzi programistycznych; systematyczny zbiór powszechnie znanych pomysłów i idei; | | |


## Wstęp do bieżącego numeru:

W tym numerze wrześniowym Redakcja publikuje pierwsze szkice z kursów algebry i analizy matematycznej w oparciu o książki S.Lang'a „Algebra" i G.M. Fichtenholz'a „Rachunek różniczkowy i całkowy" oraz pierwsze sprawozdania z kursów poprzez języki programowania Python oraz Java.

## Spis aktualnie rozwijanej zawartości:

# Kurs programowania poprzez język Python.

## Sprawozdanie 1.

Jest rok 2023. Nie można nauczyć się programować bez papierowej książki w ręku. Książek jest mnóstwo, Redakcja wybór pozostawia czytelnikowi. Po wyszukaniu hasła "Python tutorial" na pierwszej stronie wyszukiwarki internetowej, odrzuceniu nie mających czytelnego spisu treści lub tłumaczenia na język polski wyników wyszukiwania, Redakcja wybrała dwa kursy do nauki, których adresy są zamieszczone poniżej. W starych wynikach wyszukiwania korzystania Redakcji z internetu odnalazł się jeszcze jeden adres ze spisem treści zaawansowanych treści. Również zamieszczony poniżej. Redakcja postanowiła dokładnie przeczytać kurs z dokumentacji, ponieważ w ten sam sposób uczyła się programować w innych językach.

Języki programowania służą do tego, by dawać rozkazy robotom.

Rozdziały 1-3. Wygląda na to, że język Python jest popularny z powodu SPL ("Standard Python Library"). Biblioteka jest zestawem umiejętności robota. Programiści języka Python mogą do tej biblioteki dodawać własne przez siebie stworzone umiejętności - moduły. Język Python pozwala łatwiej wyrażać skomplikowane instrukcje niż inne języki programowania w początkach XX wieku: powiedzieć więcej w małej ilości słów. Podobno łatwo go wbudować do urządzeń tak by można je było programować: na przykład napisać program do włączenia światła i czajnika do zaparzania herbaty zaraz po zadziałaniu porannego budzika.

Po przejrzeniu wstępu do języka Python Redakcji udało się: obliczyć ilość sekund w XIX i w XX wieku (ponad trzy miliardy i sto milionów), skorzystać z napisów (string), skorzystać z list (list); były też błędy.

https://docs.python.org/3/tutorial/
https://www.learnpython.org/pl/
https://fullstackpython.com

# Notatka 1. O organizacjach matematycznych na Ziemi.

Redakcja postanowiła zrobić listę towarzystw, kół i związków matematyków. Po przejrzeniu kilku początkowych stron z wynikami wyszukiwania w przeglądarce internetowej Redakcja znalazła stronę internetową z przygotowaną listą takich organizacji. Redakcja wyświetliła strony Towarzystwa Matematycznego Republiki Chińskiej oraz Japońskiego Towarzystwa Matematycznego:

https://www.cimpa.info/en/node/64#submenu-1
https://www.cms.org.cn/
https://www.mathsoc.jp/

# Kurs programowania poprzez język Java.

## Sprawozdanie 1.

Podobnie do kursu języka Python (.2023) w tym czasopiśmie Redakcja przejrzała wyniki z pierwszej strony wyszukiwarki internetowej dla hasła "Java tutorial" i pomijając te, które nie miały czytelnych spisów treści, wybrała dwa kursy: pierwszy - na stronie oracle.com - i drugi - na stronie geeksforgeeks.org . Kursy zamieszczone na stronach w3schools.com, tutorialspoint.com i javatpoint.com są uproszczone: dla osób, które programują, a chcą przejrzeć strukturę języka lub przypomnieć fragment kodu do użycia. Kurs na stronie learn.microsoft.com ma spis treści, ale jest to plan na kilka lat lub miesięcy, ścieżka zawierająca linki do wysoko płatnych treści, co nie oznacza że nie warto z nich korzystać, ponieważ według Redakcji - nie da się nauczyć programowania bez papierowej książki w ręku.

W historii internetowej Redakcji jest dużo linków do stron dotyczących Javy - Redakcja przejrzy tę historię do końca teraźniejszego miesiąca.

Kurs umieszczony na stronie oracle.com Redakcja czytała już kilkukrotnie (bez tworzenia graficznego interfejsu użytkownika (GUI - Graphical User Interface) i niektórych treści zaawansowanych). Kurs ten był napisany dla wersji ósmej języka Java, ale w przyjemnej swojej formie jest łatwo przyswajalny.

Kurs w tym czasopiśmie obejmie pierwsze pięć punktów z rozdziału "Trails Covering the Basics" raczej z ich uproszczeniem do wykorzystania w środowisku do wizualizacji artystycznej Processing. Podrozdziały "Deployment" i o egzaminie są dla specjalistów - korzystając z zaawansowanego środowiska programistycznego (IDE: Integrated Development Environment) jak n.p. Eclipse prawie nie widzi się plików z rozszerzeniem .class czy .jar . Programista wprowadza program i naciśnięciem przycisku w skonfigurowanym środowisku programistycznym uruchamia testowanie i wysyła go do miejsc, w których będzie uruchomiony.

Artykuł o znaczeniu na pewno nie tylko historycznym - napisany przez twórców języka: https://www.oracle.com/java/technologies/introduction-to-java.html - Redakcja przy tym czytaniu pomija. Język Java jest tłumaczony (kompilowany - compile) (?) do plików .class grupowanych w katalogi .jar, które mogą być uruchamiane na platformach (platform), na których znajduje się maszyna wirtualna Javy (JVM) (architecture neutral, portable). To jest opisane w pierwszym rozdziale "Overview of Java" w kursie na stronie geeksforgeeks.org . JVM jest maszyną-robotem, który wykonuje kod bajtowy (byte code) (WORA: write once run anywhere), który powstaje automatycznie w wyniku kompilacji programu w języku Java; i nie tylko: prawdopodobnie dowolny język może być kompilowany do kodu bajtowego (również język Python). Tworzeniem maszyn wirtualnych Javy działających na platformach zajmują się specjaliści. Powodów dla których firma Google zamieniła w środowisku Android Studio język Java na język Kotlin Redakcja nie zna (robust: łatwy przy tworzeniu oprogramowania, czytelny i łatwy do zrozumienia dla ludzi, bardziej zwięzły, część programu jest zarządzana automatycznie: w Javie n.p. garbage collection; high performance) (?).

"Buzzword'y" w informacjach o oporogramowaniu są celami (paradygmatami: cechami języków, narzędzi, instytucji), które osiągnąć próbują twórcy.

Język Java nie jest prosty (Simple). Jest zorientowany obiektowo (object oriented), zorientowany wątkowo (thread oriented), rozproszony (distributed oriented). Według Redakcji to jego semantyka i biblioteki. JVM wykonuje program, JRE zawiera JVM i podstawowe biblioteki z algorytmami, JDK zawiera JRE oraz narzędzia do transformacji programów (development tools). W IDE-ach są wbudowane JDK. Przykładami narzędzi mogą być rozszerzenia IDE (extensions; wtyczki - plugins), które pomagają automatycznie testować programy, generować czytelnie wyglądającą ich dokumentację, umieszczać programy w miejscach gdzie będą działać, generować szkielet do wypełniania kodem. Prawdopodobnie każde IDE jest zbudowane modułowo (module: szuflada w skrzynce narzędziowej) i może mieć modyfikowaną funkcjonalność (rozszerzenia - extensions, wtyczki - plugins).

Biblioteka jest podręczną skrzynką narzędziową (toolkit), API jest pomocą samochodową do której trzeba zadzwonić. Różnica pomiędzy biblioteką, API a narzędziem jest niewielka: po jedne trzeba sięgnąć, do innych zadzwonić.

W środowisku Processing niewiele z tych elementów występuje. Wszystko jest uproszczone. Po przejrzeniu przykładów dołączonych do środowiska Processing Redakcja wyświetliła kilka białych kółek na ekranie.

https://docs.oracle.com/javase/tutorial/
https://www.geeksforgeeks.org/java/
https://learn.microsoft.com/en-us/azure/developer/java/learning-resources/fundamentals
https://www.w3schools.com/java/
https://www.tutorialspoint.com/java/index.htm
https://www.javatpoint.com/java-tutorial
Przykłady IDE: Processing, Microsoft Visual Studio Code, Sublime, Eclipse, JetBrains IntelliJ IDEA, Apache NetBeans, Oracle JDeveloper, Google Android Studio, ... (?)

# Kurs analizy matematycznej.

## Szkic 1. ::GMFI-RI::

Szkice kursu analizy matematycznej opartego o książkę G.M.Fichtenholz'a (Григорий Михайлович Фихтенголь) "Rachunek różniczkowy i całkowy" będące publikowane w tym czasopiśmie obejmą części tomu I z uwzględnieniem części tomu II. Należałoby tom II z uwzględnieniem tomu I. Początkowe rozdziały tomu I są powtórką materiału polskiej szkoły średniej końca XX wieku. Cała książka wydaje się być sumą analizy matematycznej XIX wieku: zebraniem oraz przeorganizowaniem treści listów oraz artykułów matematyków według występujących pojęć i teorii. Czytając Redakcja radzi przeorganizować tworzone notatki według funkcji i wzorów w niej występujących w co najmniej stukartkowym zeszycie formatu B5 koniecznie z twardą okładką. Taa książka nie nadaje się do czytania w tramwaju czy w pociągu. Raczej należy ponudzić się w domu, mieszkaniu, akademiku lub na kocu pod parasolem na łące, a najlepiej przy dużej białej suchościeralnej tablicy: inaczej powtarzanie definicji liczb wymiernych i rzeczywistych, granic oraz ciągłości według Heine'go (Heinrich Eduard Heine) oraz Cauchy'ego (Augustin Louis Cauchy) spowoduje niechęć i przeskakiwanie kroków obliczeń podczas czytania dalszych rozdziałów. Notując - grupując informacje - Redakcja radzi zapisywać odnośniki do tomu, rozdziału, paragrafu i strony, na której znajdują się rozproszone a zbierane w trakcie czytania informacje o funkcjach: stworzyć własny format takich odnośników (n.p. ::GMFII-RVIII-§5-str.70-p.290::).

::GMFI-RI:: to lista tematów wprowadzających liczby rzeczywiste:

liczby rzeczywiste
działania
aksjomat Archimedesa (Archimedes)
przekroje Dedekinda (Julius Wilhelm Richard Dedekind)
dobre uporządkowanie
gęstość
rozwinięcia na ułamki
kresy zbiorów
działania i ich zgodność z porządkiem
wartość bezwzględna
pierwiastki i potęgi
logarytmy
niewymierność

Czytając można zastanowić się nad aksjomatami spełnianymi przez te liczby oraz ich wartością bezwzględną, szukać innych typów liczb, ich wartości bezwzględnych oraz topologii. To zadanie jest trudne; szkic rozwiązania znajduje się w ::SLA-RXII::.

Według Redakcji GMF to dobry, przekrojowy, okrągły i doszczętny wstęp do podstaw matematycznych (wszystkich ?) teorii (fizycznych, chemicznych, ekonomicznych ?) opartych o pojecie nieskończenie małej. Książka nie przedstawia uogólnionego pojęcia przestrzeni Banacha (Stefan Banach), co jest i plusem i minusem tej książki. Taki wykład trudno połączyć z obliczeniami wzorów funkcji specjalnych, jest skomplikowany dla osób po raz pierwszy czytających książkę o analizie matematycznej, ale ułatwia zrozumienie ogólnych pojęć. Niestety, najlepszych znanych Redakcji wykładów z teorii przestrzeni Banacha oraz teorii miary Redakcja nie widzi wydrukowanych.

Książka G.M.Fichtenholz'a jest książką kucharską XIX-wiecznej matematyki.

# Kurs algebry.

## Szkic 1. ::SLA-RVII::

Książka Serg'a Lang'a (Serge Lang) "Algebra" jest trudna, ale jest baardzo satysfakcjonująca. Zaskoczeniem jest ilość ich zdefiniowanych typów z wydaje się bardzo prostego pojęcia "grupy" (group): zestawu elementów, które można parami mnożyć. Czym różnią się dodawanie i mnożenie? Czy ta książka jest opisem wszystkich możliwych do wyobrażenia zestawów liczb? Skoro istnieją liczby uproszczone, to czy liczby znane ze szkoły są uproszczeniem? I czy prawdziwa jest hipoteza:

Każda algebra jest komutatywna. (?) (!)

Książka podzielona na trzy rozdziały dotyczące teorii grup, teorii Galois (Évariste Galois), algebry liniowej. W związku z bardzo zwięzłym spisem treści książki Redakcja postanowiła w tym roku, w tym czasopiśmie publikować dokładniejsze spisy treści, fragmenty notatek oraz wskazówki do paragrafów rozdziału drugiego dotyczącego klasyfikacji zestawów liczb, w których mnożenie jest przemienne (commutative): teorii Galois.

Struktura przemiennych pierścieni, ciał i modułów (module) jest przedmiotem algebry przemiennej stosowanym w geometrii algebraicznej (algebraic geometry). (?) (!)

Rozdział VII. Rozszerzenia algebraiczne.

Rozdział służy wprowadzeniu pojęcia ciała (field): zestawu liczb podlegających przemiennym i łącznym działaniom dodawania oraz mnożenia - rozdzielnych względem siebie - wraz z 1 i 0 oraz z możliwością odwracania elementów (... bez dzielenia przez 0). Przedstawiona klasyfikacja takich zestawów polega na wyznaczeniu wszystkich ciał prostych (prime field): albo 1 + ... + 1 = 0 (charakterystyka = 0) albo nie (liczby wymierne znane ze szkoły) oraz ich skończonych rozszerzeń (finite extensions). Ciałami pierwszego typu są {0,1,2,...,p} dla liczb pierwszych p. Jeśli p nie jest liczbą pierwszą, wtedy wynikiem nietrywialnego mnożenia może być 0: 4 * 5 mod 10 = 0. Ze względu na tą cykliczność takie zestawy liczb nazwane były pierścieniami (ring). Wiele pojęć matematycznych ma nietrafne nazwy, ponieważ te zostały wprowadzone dla prostszych obiektów przed poznaniem ich uogólnień.

§1: wielomiany nierozkładalne (irreducible polynomials) i ich pierwiastki (zeros), rozszerzenia ciał i ich automorfizmy, rozszerzenia skończone, algebraiczne oraz skończenie generowane (finitely generated).

Motywem przewodnim paragrafu jest pojęcie wyróżnionej klasy rozszerzeń w prosty sposób reprezentowalnej na diagramach. Rozszerzenia skończenie generowane nie tworzą wyróżnionej klasy rozszerzeń. Pomimo dużej ilości z premedytacją wstawionych literówek w posiadanych przez Redakcję wydaniach diagramy w książce są bardzo ważne i należy je zrozumieć tak jak są podane: bez uproszczeń przy trudnościach w zrozumieniu, bez wymiany monomorfizmów na izomorfizmy. Bardzo często (!) lematy są ważniejsze od twierdzeń. Elementy przestępne (transcendent) i ich rozszerzenia omówione są w Rozdziale X (::SLA-RX::).

Po przeczytaniu rozdziału Czytelnik rozumie dorzucanie nieistniejących pierwiastków wielomianu (części lub wszystkich), zależność pomiędzy algebraicznością (algebraic) liczby a skończonością jej rozszerzenia, stopniami wielomianu minimalnego i rozszerzenia ciała, tworzenie piętrowych rozszerzeń skończonych.

§2: przy pierwszym czytaniu, być może, należy pominąć czytanie treści dowodów, a czytając rozróżnić pojęcia algebraicznego domknięcia ciała oraz ciała algebraicznie domkniętego. Rozdział ten może sprawiać trudności związane z podstawami matematyki oraz aksjomatyką teorii zbiorów.

§3: rozszerzenia normalne: ich piętra działają zupełnie niezależnie pod względem grup automorfizmów ciał (trochę jak te okrągłe zamki szyfrowe w sejfach ?). Nie jest to klasa wyróżniona. Powstają przez dorzucenie wszystkich pierwiastków zestawu wielomianów.

§4, §7: opisują dwie klasy wyróżnione rozszerzeń: rozdzielcze (seperable), i radykalne (z inną czasem używaną nazwą czysto nierozdzielczych) (radical, pure inseperable). Rozszerzenia nierozdzielcze mogą zawierać elementy rozdzielcze, ale również, nierozdzielcze. Rozszerzenia radykalne nie zawierają elementów rozdzielczych. Rozszerzenie skończone jest rozdzielcze w.i.t.w., gdy wszystkie jego elementy są rozdzielcze, w.i.t.w., gdy jego stopień nierozdzielczy jest równy 1. Stopień rozdzielczy może być podzielny przez charakterystykę ciała p. Stopień rozszerzenia, stopień rozdzielczy rozszerzenia i stopień nierozdzielczy rozszerzenia są multiplikatywne. Wystarczy udowodnić dwa.

§5: o ciałach skończonych w następnym szkicu.

§6: twierdzenie o elemencie prymitywnym. (!!)

| | | | |
|---|---|---|---|
| Numer: | **MATINF 0/2023** | | |
| Data wydania: | 31 lipca 2023 | Druk: | 3 |
| Adresy Redakcji: | czasopismo.matinf@gmail.com | Witryny informacyjne: | https://github.com/czasopismo-MATINF/czasopismo-MATINF |
| Czytelnik: | osoba samodzielnie ucząca się, student | | |
| Cel: | systematyczne kursy podstaw programowania, matematyki, algebry i analizy matematycznej, artykuły o takowej tematyce; recenzje książek do nauki, narzędzi do programowania, listy zasobów informatycznych, serwisów internetowych; po dwudziestu latach wybuchu programistycznego niepotrzebna już promocja programowania, ale powrót na jego podstawowy poziom i znalezienie miejsca dla jego podstaw przy potwornie zwiększającym się stopniu skomplikowania algorytmów oraz narzędzi programistycznych; systematyczny zbiór powszechnie znanych pomysłów i idei; | | |


## Wstęp do bieżącego numeru:

W numerze wrześniowym czasopisma rozpoczną ukazywać się pierwsze części kursów programowania w językach Python i Java. Kursy te będą zawierały opisy instrukcji języków oraz obsługi dostępnych narzędzi do programowania w tych językach i zgodnie z przesłaniem MATINF: będą odpowiednie - dla osób, które samodzielnie się uczą - jako systematyczna pomoc w organizacji nauki i nabywania praktycznych umiejętności. Kurs języka Python będzie odpowiedni dla osób bez znajomości programowania, ponieważ oddelegowany pracownik redakcji też będzie uczył się tego języka, a z kursów korzystając dla powtórzenia wiadomości i sprawdzenia swoich umiejętności. Również po letnich wakacjach zaczną się pojawiać szkice z kursów analizy matematycznej i algebry z wykorzystaniem polskich, i polskich tłumaczeń znanych i popularnych podręczników.

## Spis aktualnie rozwijanej zawartości: