

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:

kurs programowania w języku Java	narzędzia programistyczne	kurs programowania w języku Python	szkice kursu z analizy matematycznej	historia matematyki przełomu XVIII i XIX wieku
listy serwisów internetowych	recenzje książek	algorytmika	szkice kursu z algebry	indeks wiadomości technicznych

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/arb01/2001>

<https://directorio.uva.es/arb01/2019>

<https://directorio.uva.es/arb01/2051>

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 premiedytacją więcej niż pięćdziesiąt dwa rozdziały, co oznacza, że czytając jeden rozdział tygodniowo, a chodząc na spacer, 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. środowisko 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 przeczytać.

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 (extensions, modules, plugins, scripts)

łatwy do wykorzystania, zrozumienia i nauczania, 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 rozdziale 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 $k(t_1, \dots, t_n)$ do elementów stałych grupy symetrycznej S_n w przykładzie czwartym wynika z rzędu tej grupy $n!$, wszystkie elementy t_1, \dots, t_n są algebraiczne nad ciałem $k(s_1, \dots, s_n)$. 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 \mathbb{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: $f_2 = X + 1$. 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 \mathbb{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 $e^{\text{Tr}} = N(e)$? 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 piętér 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:

kurs programowania w języku Java	narzędzia programistyczne	kurs programowania w języku Python	szkice kursu z analizy matematycznej	historia matematyki przełomu XVIII i XIX wieku
listy serwisów internetowych	recenzje książek	algorytmika	szkice kursu z algebry	indeks wiadomości technicznych

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 znakiem przypisać można różne algorytmy rozwijając semantykę i skracając długość programowania funkcyjnego:

```
[...] 国立研 - wykonaj kolejno operacje 国立研 na
strumieniu danych [...] (skrótowy 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 pomniejszych 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 tablicie 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 pszczoł? (!)

Przykłady Czytelnik może podać sam:

```
[* 0 - 100 : "a-r" *] - stwórz 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 niezmiennie. 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,  
(zmiennie)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 formularzy 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 obliczającą 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 bardzo wysokiej jakości.

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

Tym razem Redakcja napisała skrypt wyświetlający 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://javaconceptsoftheday.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:
<https://www.baeldung.com/tutorials/>
<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 nakładają 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 istnieją giętkie "smartfony", przezroczyste ekrany z "genetycznie" modyfikowanego plastiku (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 przerzeczyste.

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 drażony z niego przez widza? (!)

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

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

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 pięt. Programowanie strukturalne to wykorzystanie instrukcji "goto", proceduralne pozwala na porządkowanie kodu w metodach; różnica pomiędzy paradygmatami obiektywnym 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) ? (!)

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

Kurs analizy matematycznej.

Szkic 2. ::GMFI-RI::

Jeśli Czytelnik lub Czytelniczki zdecydowali kupić książkę, co najmniej stukartkowy zeszyt formatu B5 w kratkę, przynajmniej dziesięć wysokiej jakości ołówków, gumkę oraz temperówkę, oraz ... znaleźli dużą białą tablicę suchocierną 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 \mathbb{R}^n , ani ścisłego wyprowadzenia pojęć powierzchni i 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. Wniosek 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 \mathbb{R}^n . Przy granicach \liminf oraz \limsup warto zauważyć, że poniżej lub powyżej może występować nieskończenie wiele elementów. Po uwzględnieniu $-\infty$ oraz $+\infty$ 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 \mathbb{Z}_p - 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 przemennych. Współczynniki dwumienne obliczamy nad \mathbb{Z} , potem redukujemy modulo p (wyobrazić mnożenie oraz dodawanie). Gdy p jest liczbą pierwszą, wtedy dzieli współczynniki dwumienne z wyjątkiem "pierwszego" i "ostatniego":

$$(x+y)^p = x^p + y^p \text{ oraz podobnie } (x+y)^{p^n} = x^{p^n} + y^{p^n} \quad (!)$$

Dwa zera $x^q - x = 0$, $y^q - y = 0$ mają ciekawą własność: $(xy)^q = (x^q)(y^q) = xy \Rightarrow (xy)^q - xy = 0$: 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. (!)

Jedznaczność 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 : $x^{(p^n)} = x^{(p^{(m+n-m)})} = (x^{(p^m)})^{(p^{(n-m)})} = x^{(p^{(n-m)})} = \dots = x^{(p^0)} = x$ (dla $m \mid n$!) - domknięcie algebraiczne ciała Z_p . 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 Z_p , potem Z_q . 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 nieprzemienne, natomiast dodanie zależności w definicji struktury zmniejsza ilość jej rzeczywistych modeli (Q oraz Z_p). 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 w języku Java	narzędzia programistyczne	kurs programowania w języku Python	szkice kursu z analizy matematycznej	historia matematyki przełomu XVIII i XIX wieku
listy serwisów internetowych	recenzje książek	algorytmika	szkice kursu z algebry	indeks wiadomości technicznych

Kurs programowania poprzez język Python.

Sprawozdanie 1.

Jest rok 2023. Nie można nauczyć się programować bez papierowej książki w rękę. 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 kilkakrotnie (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 obejmuje 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.

<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, ... (?)

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>

Kurs analizy matematycznej.

Szkic 1. ::GMFI-RI::

Szkice kursu analizy matematycznej opartego o książkę G.M.Fichtenholz'a (Григорий Михайлович Фихтенгольц) "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 Archimedes (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 pojęcie 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 bardzo 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 przemienym 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 $1 + \dots + 1 = 0$ (charakterystyka = 0) albo nie (liczby wymierne znane ze szkoły) oraz ich skończonych rozszerzeń (finite extensions). Ciałami pierwszego typu są $\{0, 1, 2, \dots, p\}$ dla liczb pierwszych p . Jeśli p nie jest liczbą pierwszą, wtedy wynikiem nietrywialnego mnożenia może być 0: $4 * 5 \bmod 10 = 0$. Ze względu na tę 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).

Motywnym 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 pięterowych 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:

kurs programowania w języku Java	narzędzia programistyczne	kurs programowania w języku Python	szkice kursu z analizy matematycznej	historia matematyki przełomu XVIII i XIX wieku
listy serwisów internetowych	recenzje książek	algorytmika	szkice kursu z algebry	indeks wiadomości technicznych