



# Szkolenia Komputerowe





# CZĘŚĆ JAVOWA

# HTML, CSS, JS



- **HTML** (*HyperText Markup Language*) – hipertekstowy język znaczników, wykorzystywany do tworzenia dokumentów hipertekstowych. Opisuje strukturę informacji zawartych wewnątrz strony internetowej, nadając znaczenie poszczególnym fragmentom tekstu oraz osadza w tekście dokumentu obiekty.

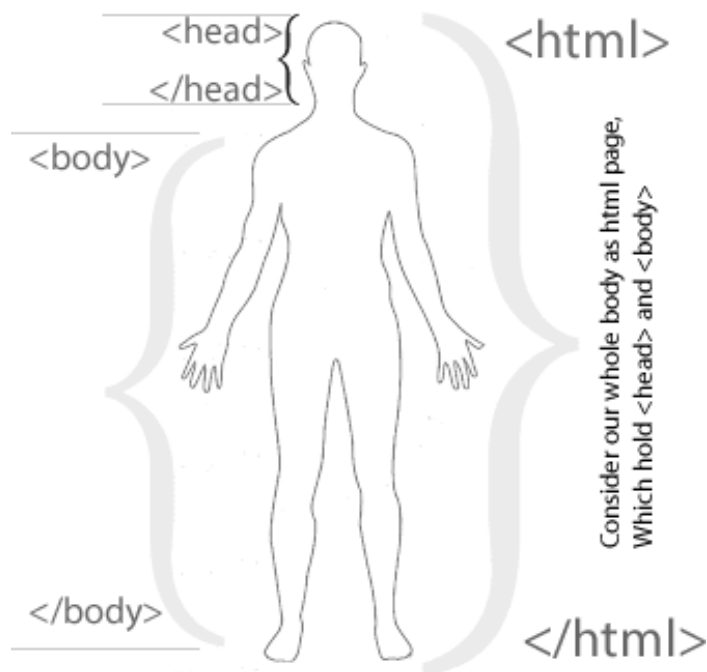
# HTML



Źródło: <https://en.wikipedia.org/wiki/HTML>



## HTML - przykłady



Źródło: <https://www.w3tweaks.com>

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Example</title>
5          <link rel="stylesheet" href="style.css">
6      </head>
7      <body>
8          <h1>
9              <a href="/">Header</a>
10         </h1>
11         <nav>
12             <a href="one/">One</a>
13             <a href="two/">Two</a>
14             <a href="three/">Three</a>
15         </nav>
```

Źródło: <https://en.wikipedia.org/wiki/HTML>

# HTML, CSS, JS



- **CSS** (*Cascading Style Sheets*) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. Arkusz stylów CSS to lista dyrektyw (tzw. reguł) ustalających w jaki sposób ma zostać wyświetlana przez przeglądarkę internetową zawartość wybranego elementu HTML. Można w ten sposób opisać wszystkie pojęcia odpowiedzialne za prezentację elementów dokumentów internetowych, takie jak rodzina czcionek, kolor tekstu, marginesy, odstęp międzywierszowy lub nawet pozycja danego elementu względem innych elementów bądź okna przeglądarki.



Źródło: <https://www.flickr.com>



## CSS - przykład

```
1  /* GENERAL STYLES
2  *-----*/
3  html, body, form, fieldset, img, img a {
4      margin: 0;
5      padding: 0;
6      border: 0;
7  }
8  body {
9      color: #414141;
10     background: url(../images/bg.jpg) repeat-x #ebe8df;
11     font-family: Arial, Helvetica, sans-serif;
12     line-height: 120%;
13     font-size: 12px;
14 }
15
16 a:link, a:visited {
17     color: #685966;
18     text-decoration: underline;
19 }
20 a:hover {
21     color: #2b212c;
22 }
23 .article_separator {
24     line-height: 5px;
25     height: 5px;
26     font-size: 5px;
27 }
28 /* SITE WIDTH
29 *-----*/
30 .rht_container {
31     width: 1020px;
32     margin: 0 auto;
33     margin-top: 25px;
34 }
```

Źródło: [https://is.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://is.wikipedia.org/wiki/Cascading_Style_Sheets)

# HTML, CSS, JS



- **JS** (*JavaScript*) – skryptowy język programowania, stworzony przez firmę Netscape, najczęściej stosowany na stronach internetowych. Skrypty te służą najczęściej do zapewnienia interakcji poprzez reagowanie na zdarzenia, walidacji danych wprowadzanych w formularzach lub tworzenia złożonych efektów wizualnych. Skrypty JavaScriptu uruchamiane przez strony internetowe mają znacznie ograniczony dostęp do komputera użytkownika.



Źródło: [https://es.wikibooks.org/wiki/Programaci%C3%B3n\\_en\\_JavaScript](https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_JavaScript)



## JS - przykład

```
function enEdition(){
    /* Ne rien faire mode edit + preload */
    if( encodeURIComponent(document.location).search(/%26preload%3D/) != -1 ) return;
    // /&preload=/

    if ( !wgPageName.match(/Discussion.*\Traduction/) ) return;
    var diff = new Array();
    var status; var pecTraduction; var pecRelecture;
    var avancementTraduction; var avancementRelecture;

    /* ***** Parser ***** */
    var params = document.location.search.substr(1, document.location.search.length).split('&');
    var i = 0;
    var tmp; var name;
    while ( i < params.length )
    {
        tmp = params[i].split('=');
        name = tmp[0];
        switch( name ) {
            case 'status':
                status = tmp[1];
                break;
            case 'pecTraduction':
```

Źródło: [https://fr.wikipedia.org/wiki/Langage\\_de\\_programmation](https://fr.wikipedia.org/wiki/Langage_de_programmation)



# HTML, CSS, JS

HTML + CSS + JS

The screenshot shows the Yahoo! homepage layout. At the top is the Yahoo! logo and a search bar. Below the logo are navigation links: Mail, News, Finance, Sports, Politics, Entertainment, Lifestyle, and More... A UNICEF banner is visible. The main content area features a large article about a car crash with the headline "After crash, toddlers 'went through pure hell'", followed by a row of smaller news snippets. To the right, there's a "Trending Now" list and a "Pre-register 2nd China International Import Expo" section. Below the main article, there's a "Gdansk, Pomeranian" section with a weather forecast and a "Scoreboard" for the MLB. The bottom of the page shows more news articles, including one about a suspect in a murder case and another about a scientific discovery.

**YAHOO!**

Mail News Finance Sports Politics Entertainment Lifestyle More...

unicef

**After crash, toddlers 'went through pure hell'**

Two brothers, ages 1 and 3, survive for several days in mangled wreckage in a ravine after a car crash kills their mother. 'It is completely amazing' +

1215 people reacting

**Trending Now**

1. Millie Bobby Brown
2. Aaron Rodgers
3. Gemma Chan
4. Lana Del Rey
5. Travel Insurance
6. 2018 Acura Rlx
7. Real Madrid
8. Student Loans
9. Dexter Fowler
10. Jamie Foxx

**Pre-register 2nd China International Import Expo**

China International Import Expo

**Gdansk, Pomeranian**

Today Thu Fri Sat

74° 43° 79° 55° 79° 61° 72° 57°

**Scoreboard** MLB

Yesterday	Today	Tomorrow
Baltimore	0	Top 3
Toronto	0	Top 3
Minnesota	Wed, Aug. 22, 8:10 PM CEST	Top 3
Chi White Sox	CSC, FSN	Top 3
Cincinnati	Wed, Aug. 22, 8:10 PM CEST	Top 3
Milwaukee	FSN, FSN	Top 3
Texas	Wed, Aug. 22, 9:35 PM CEST	Top 3
Oakland	Wed, Aug. 22, 10:10 PM CEST	Top 3
Houston	Wed, Aug. 22, 10:10 PM CEST	Top 3
Seattle	FSN	Top 3
Philadelphia	Thu, Aug. 23, 1:05 AM CEST	Top 3
Washington	MASS, ESP	Top 3

**More scores >**

**Lifestyle** Yahoo Lifestyle

**'I shouldn't have worn the green shirt': Mom finds out that schools use greenscreens for class photos after it's too late**

Laure Boone Hubsell recommended that her son Carter wear a green shirt, so that it would really pop against any gray or taupe background the school would be using.

**Science** Fox News

**Speedy flip in Earth's magnetic field could cause trillions in damage, scientists warn**

A new study of previous reversals of Earth's magnetic field has found a rapid shift occurred within two centuries - a discovery that has prompted researchers to warn of a...

**U.S.** Good Morning America

**Suspect in Mollie Tibbetts' murder passed background check despite immigration status, employer says**

Yarrabee Farms, the Iowa farm that employed the suspect charged with killing Mollie Tibbetts, defended its vetting of Cristian Bahena Rivera late Tuesday. The farm says Rivera passed t...

**U.S.** The Des Moines Register

**'You think small towns are safe': UI students shocked at discovery of body in Mollie Tibbetts' search**

**Mollie Tibbetts told man who followed her she would call police**

WISN

Źródło: <https://www.yahoo.com/>

# GIT – System Kontroli Wersji

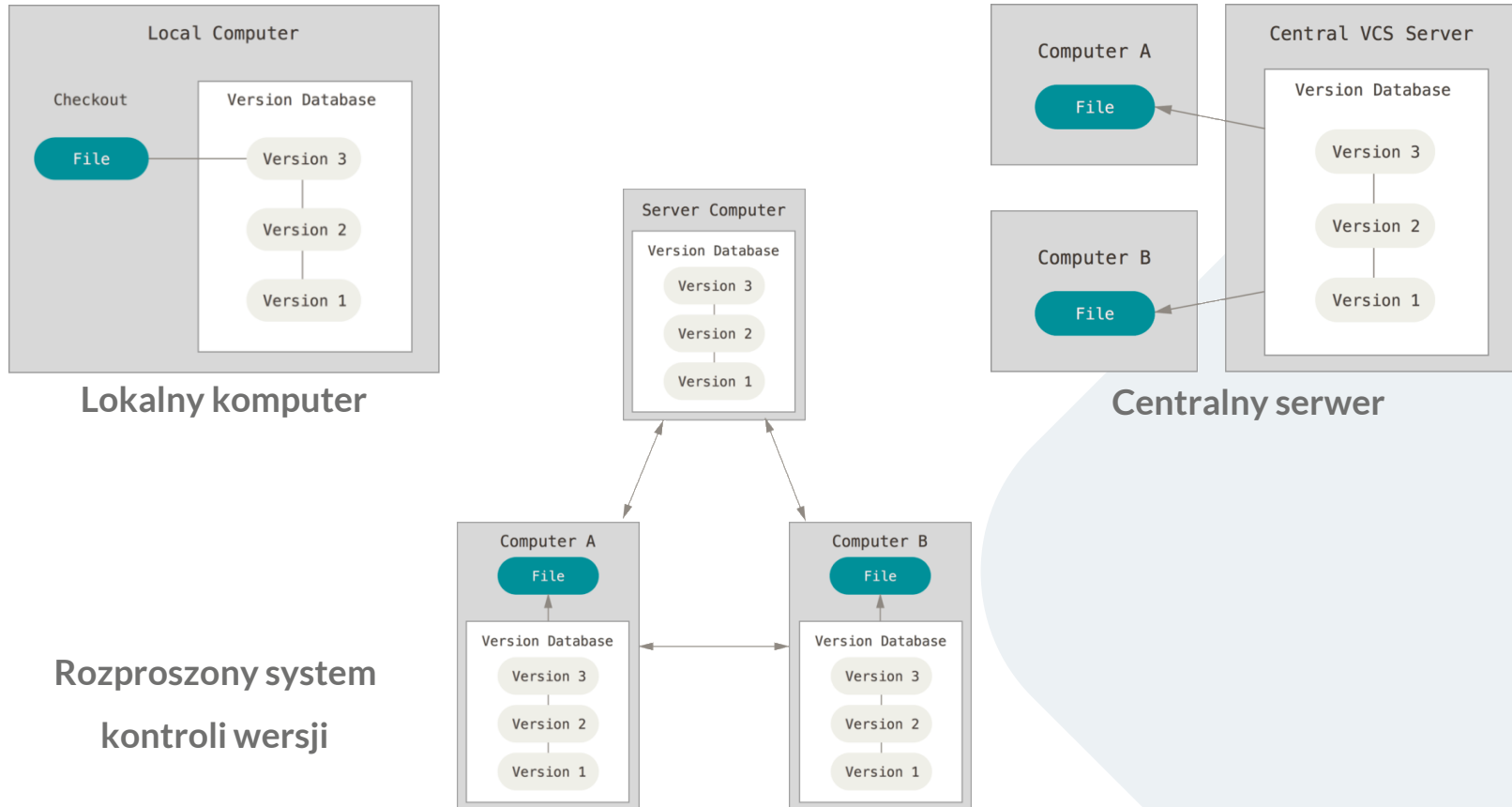


- **Git** – rozproszony system kontroli wersji. Stworzył go Linus Torvalds jako narzędzie wspomagające rozwój jądra Linux. Git stanowi wolne oprogramowanie i został opublikowany na licencji GNU GPL w wersji 2. Pierwsza wersja narzędzia Git została wydana 7 kwietnia 2005 roku, by zastąpić poprzednio używany w rozwoju Linuksa, niebędący wolnym oprogramowaniem, system kontroli wersji BitKeeper.



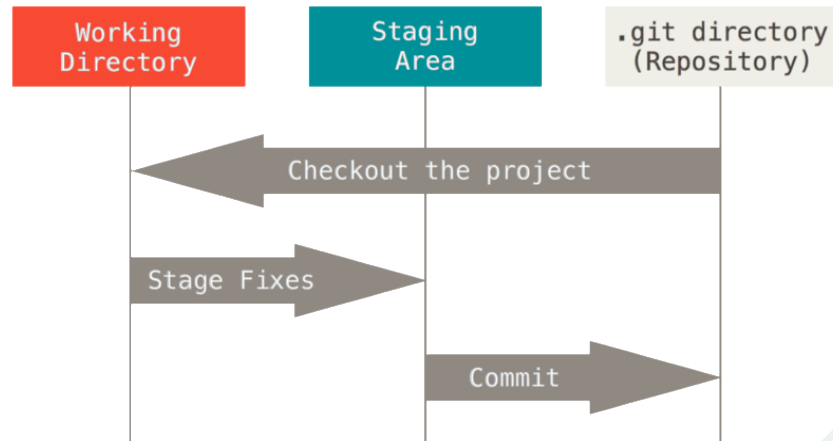
Źródło: <https://git-scm.com/>

# GIT – System Kontroli Wersji

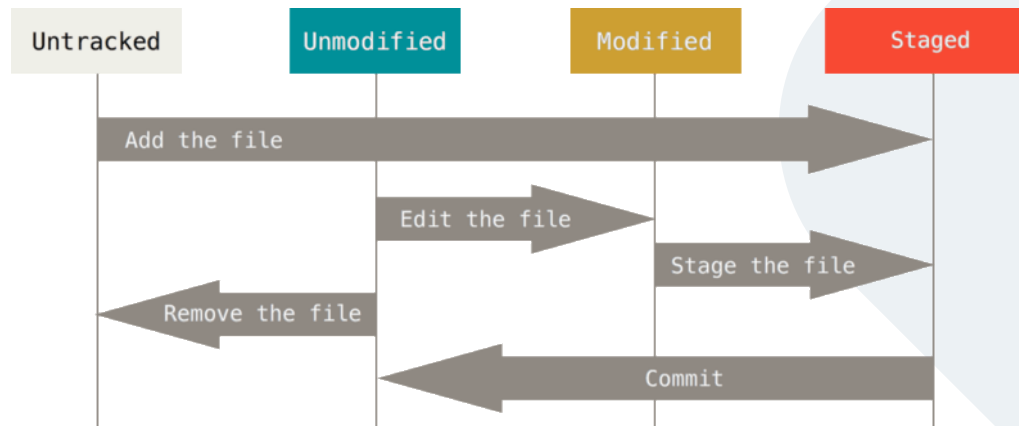


Źródło: <https://git-scm.com/>

# GIT – System Kontroli Wersji



## Praca z Gitem

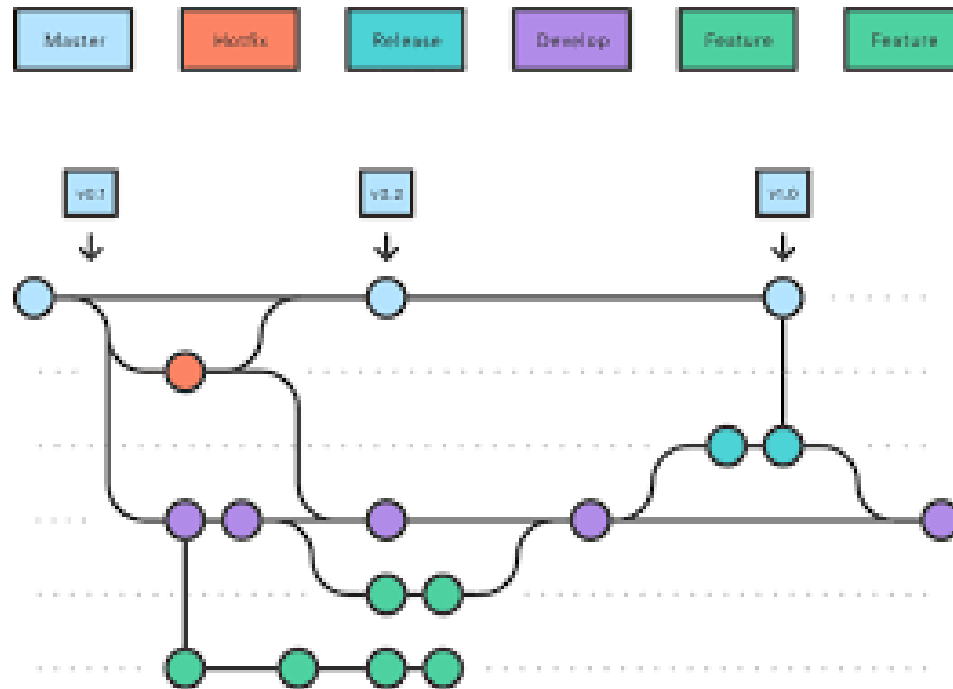


Źródło: <https://git-scm.com/>

# GIT – System Kontroli Wersji



GIT Flow



Źródło: <https://www.atlassian.com>

# Java – podstawy i zaawansowana



- Zaprojektowana przez Jamesa Goslinga
- Rozwijana przez Sun Microsystems jako język do programowania urządzeń elektronicznych
- W 1995 roku zadebiutowała jako język do tworzenia aplikacji dla Internetu
- Wzrost popularności dzięki dostępności przeglądarki Netscape
- Język ogólnego przeznaczenia
- Przenośny, działający na wielu platformach
- Kod źródłowy kompilowany do bytecode'u
- Programy uruchamiane w Java Virtual Machine (JVM)
- WORA - Write Once, Run Anywhere



Źródło: <https://www.redbubble.com>

# Java – podstawy i zaawansowana

---



- **JDK** - Java Development Kit
  - sposób instalacji zależny od systemu operacyjnego
  - można pobrać ze strony Oracle lub OpenJDK
  - ustawiamy zmienną środowiskową JAVA\_HOME
- **JRE** - Java Runtime Environment
  - instalujemy oddzielnie jeśli nie mamy JDK (które posiada JRE)
- **IDE** - Integrated Development Environment
  - darmowe: Eclipse, NetBeans, IntelliJ IDEA Community
  - płatne: IntelliJ IDEA Ultimate

# Java – podstawy i zaawansowana

---



- **Programowanie obiektowe** (*Object-Oriented Programming, OOP*) – paradygmat programowania, w którym programy definiuje się za pomocą obiektów – elementów łączących stan (czyli dane, nazywane najczęściej polami) i zachowanie (czyli procedury, tu: metody). Obiektowy program komputerowy wyrażony jest jako zbiór takich obiektów, komunikujących się pomiędzy sobą w celu wykonywania zadań.
- Podejście to różni się od tradycyjnego programowania proceduralnego, gdzie dane i procedury nie są ze sobą bezpośrednio związane. Programowanie obiektowe ma ułatwić pisanie, konserwację i wielokrotne użycie programów lub ich fragmentów.
- Największym atutem programowania, projektowania oraz analizy obiektowej jest zgodność takiego podejścia z rzeczywistością – mózg ludzki jest w naturalny sposób najlepiej przystosowany do takiego podejścia przy przetwarzaniu informacji.



# Java – podstawy i zaawansowana

---



- **kod źródłowy** - zapis programu komputerowego przy pomocy określonego języka programowania, opisujący operacje jakie powinien wykonać komputer na zgromadzonych lub otrzymanych danych. Kod źródłowy jest wynikiem pracy programisty i pozwala wyrazić w czytelnej dla człowieka formie strukturę oraz działanie programu komputerowego.
- **kompilator** - program, który kod źródłowy zamienia w kod napisany w innym języku. Oprócz tego kompilator ma za zadanie odnaleźć błędy leksykalne i semantyczne oraz dokonać optymalizacji kodu.
- **kod bajtowy** – (*bytecode*), wynik kompilacji programu napisanego w Javie, kod ten jest zrozumiały dla środowiska uruchomieniowego Java (JVM).



## Pierwszy program

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

# Java – podstawy i zaawansowana



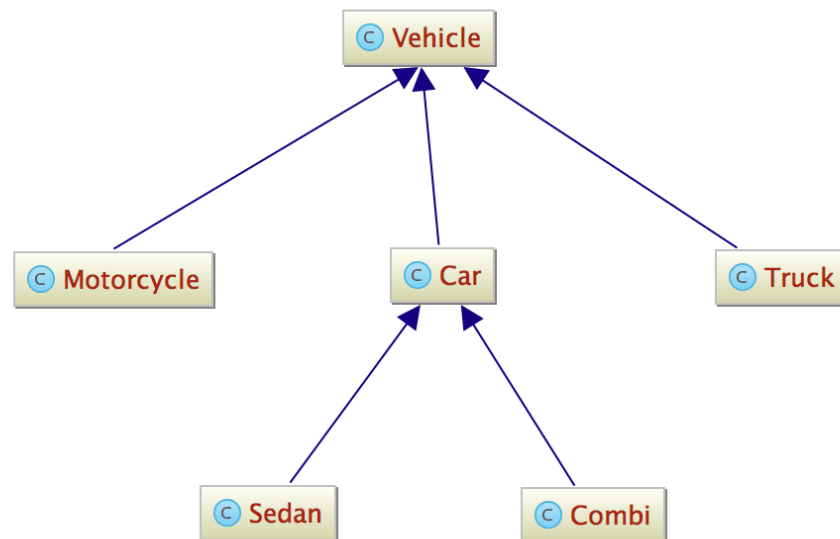
- **klasa** - wzorzec określający typ
- **obiekt** - instancja klasy
- **stan** - zmienne instancyjne
- **zachowanie** - metody

```
public class UserDTO {  
    private String username;  
    private String fullName;  
    private String role;  
    private boolean active;  
  
    public UserDTO() {  
        super();  
    }  
  
    public String getUsername() {  
        return username;  
    }  
  
    public void setUsername(String username) {  
        this.username = username;  
    }  
  
    public String getFullName() {  
        return fullName;  
    }  
  
    public void setFullName(String fullName) {  
        this.fullName = fullName;  
    }  
  
    public String getRole() {  
        return role;  
    }  
  
    public void setRole(String role) {  
        this.role = role;  
    }  
  
    public boolean getActive() {  
        return active;  
    }  
  
    public void setActive(boolean active) {  
        this.active = active;  
    }  
}
```

# Java – podstawy i zaawansowana



- **Dziedziczenie** jest relacją pomiędzy bardziej ogólną klasą (nadklasą) a bardziej konkretną klasą (podklasą)
- Podklasa dziedziczy stan (pola) oraz zachowanie (metody) od nadklasy
- Aby wyrazić dziedziczenie używamy słowa kluczowego `extends`
- Klasa `Object` - korzeń hierarchii
- Relacja **Is-A**
  - **Car Is-A Vehicle**
  - **Sedan Is-A Car**



# Java – podstawy i zaawansowana



- **Kompozycja** - tworzenie nowej klasy na podstawie innych klas
- Relacja **Has-A**
  - Car **Has-A** Engine



# Java – podstawy i zaawansowana



- **Abstrakcją** w programowaniu nazywamy pewnego rodzaju uproszczenie rozpatrywanego problemu, polegające na ograniczeniu zakresu cech manipulowanych obiektów wyłącznie do cech kluczowych dla algorytmu, a jednocześnie niezależnych od implementacji. W tym sensie abstrakcja jest odmianą formalizmu matematycznego.
- Języki programowania (w szczególności języki obiektowe) mogą oferować specjalne narzędzia do tworzenia pewnych typów abstrakcji, jednak pojęcie to nie zależy w żaden sposób od konkretnego języka.
- Każde uogólnienie problemu mające na celu wyizolowanie jego kluczowych aspektów jest abstrakcją.
- Termin abstrakcja bywa również używany w informatyce na określenie pokrewnej, jednak nie tożsamej techniki ukrywania implementacji za wspólnym interfejsem, który może korzystać z wielu różnych bibliotek do osiągnięcia pożądanej funkcjonalności. W tym sensie można mówić o abstrakcji bazy danych, mając na myśli kod który dostosowuje interfejs danej bazy danych do wspólnej postaci oczekiwanej przez program.

# Java – podstawy i zaawansowana

---



- **Hermetyzacja** (enkapsulacja) – jedno z założeń programowania obiektowego. Hermetyzacja polega na ukrywaniu pewnych danych składowych lub metod obiektów danej klasy tak, aby były one dostępne tylko metodom wewnętrznym danej klasy lub funkcjom zaprzyjaźnionym.
- Gdy dostęp do wszystkich pól danej klasy jest możliwy wyłącznie poprzez metody, lub inaczej mówiąc: gdy wszystkie pola w klasie znajdują się w sekcji prywatnej lub chronionej, to taką hermetyzację nazywa się hermetyzacją pełną.
- **Polimorfizm** (z gr. wielopostaciowość) - mechanizmy pozwalające programiście używać wartości, zmiennych i podprogramów na kilka różnych sposobów. Inaczej mówiąc jest to możliwość wyabstrahowania wyrażen od konkretnych typów.

# Java – podstawy i zaawansowana

---



- Wiele mechanizmów polimorficznych można napisać ręcznie, jednak wiąże się to często z koniecznością powielania kodu z jedynie niewielkimi poprawkami, a co za tym idzie rozrost kodu źródłowego i jego zaciemnienie.
- Istotą polimorfizmu jest to aby to system decydował o szczegółach, nie programista. Przez system należy tu rozumieć kompilator i system czasu wykonania.
- Niektóre decyzje mogą być podjęte już na etapie kompilacji, mamy wtedy do czynienia z polimorfizmem statycznym (**czasu kompilacji**).
- Czasami jednak decyzja musi zostać odłożona do momentu wykonywania programu - polimorfizm dynamiczny (**czasu wykonania**).



# Java – podstawy i zaawansowana

---



```
Animal cat = new Cat("Stevens");
```

- **Zmienna referencyjna** może mieć tylko jeden typ i nie można go zmienić
- **Typ referencji** (zmiennej) określa jakie metody mogą być na niej wywoływane
- **Typ obiektu** określa, która przestłonięta metoda metoda zostanie wywołana
- **Typ referencji** określa, która przeciążona metoda zostanie wywołana

# Java – programowanie podstawowe i zaawansowane

---



- **Algorytm** – skończony ciąg jasno zdefiniowanych czynności, koniecznych do wykonania pewnego rodzaju zadań. Sposób postępowania prowadzący do rozwiązania problemu.
- **Teoria złożoności obliczeniowej** – dział teorii obliczeń, którego głównym celem jest określanie ilości zasobów potrzebnych do rozwiązania problemów obliczeniowych. Rozważanymi zasobami są takie wielkości jak czas, pamięć lub liczba procesorów.
- **Struktura danych** (*data structure*) – sposób przechowywania danych w pamięci komputera. Na strukturach danych operują algorytmy.

# Java – programowanie podstawowe i zaawansowane



## Kolejka FIFO – First In First Out



Źródło: <http://www.wvxu.org/post/did-great-recession-bring-back-1930s>

# Java – programowanie podstawowe i zaawansowane



## Kolejka LIFO – Last In First Out



Źródło: <https://www.pexels.com/photo/plate-lunch-meal-stack-95218/>

# Java – programowanie podstawowe i zaawansowane



## Kolejka priorytetowa

<b>CZERWONY</b>	● pomoc natychmiastowa
<b>POMARAŃCZOWY</b>	● czas oczekiwania do 10 min.
<b>ŻÓŁTY</b>	● czas oczekiwania do 1 godz.
<b>ZIELONY</b>	● czas oczekiwania do 2 godz.
<b>NIEBIESKI</b>	● czas oczekiwania do 4 godz.

Źródło: <https://pabianice.tv/czerwony-pacjent-ma-pierwszenstwo-co-z-innymi/>



Źródło: <http://www.ratownik-med.pl/aktualnosci/11373-zastrzyk-pieniedzy-dla-sor-ow.html>

# Java – programowanie podstawowe i zaawansowane

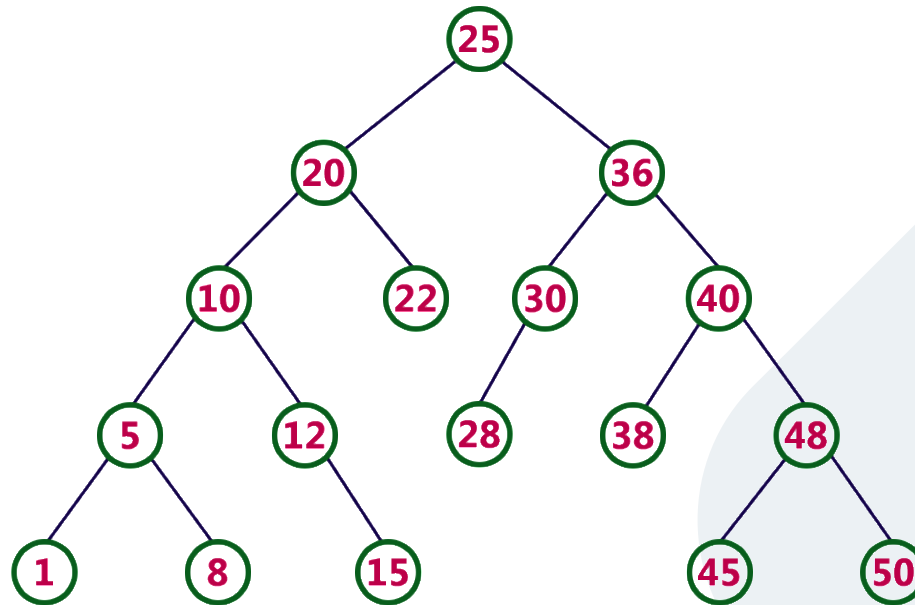


- **Drzewo** – struktura danych reprezentująca drzewo matematyczne. W naturalny sposób reprezentuje hierarchię danych (obiektów fizycznych i abstrakcyjnych, pojęć, itp.) jest więc stosowane głównie do tego celu. Drzewa ułatwiają i przyspieszają wyszukiwanie, a także pozwalają w łatwy sposób operować na posortowanych danych.
- Znaczenie tych struktur jest bardzo duże i ze względu na swoje własności drzewa są stosowane praktycznie w każdej dziedzinie informatyki (np. bazy danych, grafika komputerowa, przetwarzanie tekstu, telekomunikacja).
- **Drzewo binarne** – drzewo, w którym stopień każdego wierzchołka jest nie większy od 3.
- W informatyce drzewo binarne to jeden z rodzajów drzewa (struktury danych), w którym liczba synów każdego wierzchołka wynosi nie więcej niż dwa. Wyróżnia się wtedy lewego syna i prawego syna danego wierzchołka.
- Drzewo binarne, w którym liczba synów każdego wierzchołka wynosi albo zero albo dwa, nazywane jest drzewem regularnym.

# Java – programowanie podstawowe i zaawansowane



Drzewo binarne



# Java – programowanie podstawowe i zaawansowane



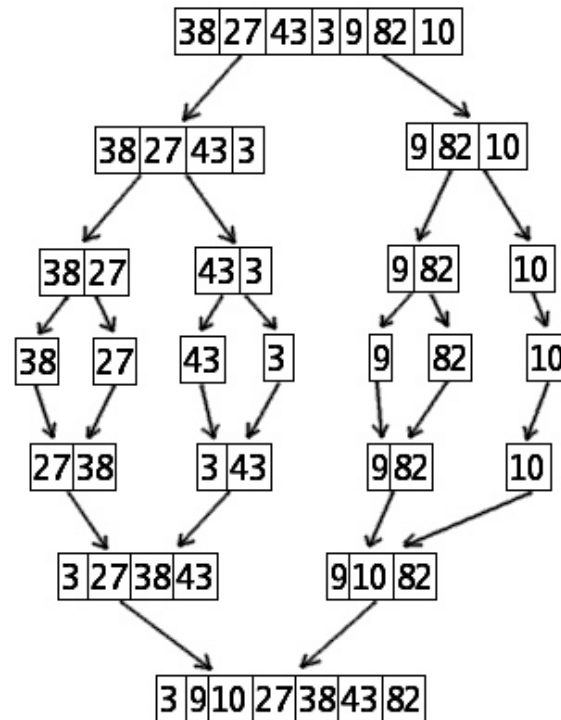
- **Sortowanie** – jeden z podstawowych problemów informatyki, polegający na uporządkowaniu zbioru danych względem pewnych cech charakterystycznych każdego elementu tego zbioru. Szczególnym przypadkiem jest sortowanie względem wartości każdego elementu, np. sortowanie liczb, słów itp.
- Algorytmy sortowania są stosowane w celu uporządkowania danych, umożliwienia stosowania wydajniejszych algorytmów (np. wyszukiwania) i prezentacji danych w sposób czytelniejszy dla człowieka.
- Jeśli jest konieczne posortowanie zbioru większego niż wielkość dostępnej pamięci, stosuje się algorytmy sortowania zewnętrznego.
- Algorytmy, do działania których nie jest potrzebna większa niż stała pamięć dodatkowa (elementy sortowane przechowywane są przez cały czas w tablicy wejściowej), nazywane są algorytmami działającymi w miejscu.
- Algorytmy sortujące, które dla elementów o tej samej wartości zachowują w tablicy końcowej kolejność tablicy wejściowej, nazywamy algorytmami stabilnymi.



# Java – programowanie podstawowe i zaawansowane



## Sortowanie

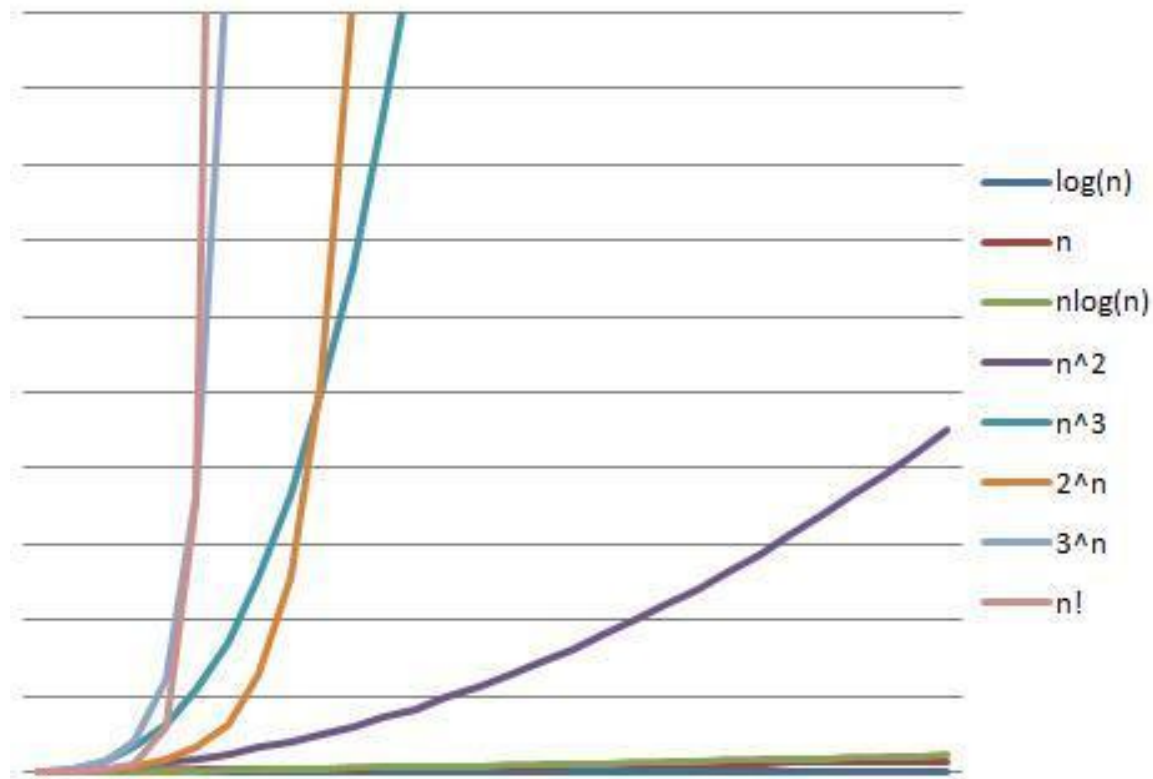


Źródło: <https://pl.wikipedia.org/wiki/Sortowanie>

# Java – programowanie podstawowe i zaawansowane



Złożoność obliczeniowa



Źródło: <http://www.algorytm.org/kurs-algorytmiki/zlozonosc-obliczeniowa.html>

# Testowanie oprogramowania i TDD



- **Testowanie oprogramowania** – proces związany z wytwarzaniem oprogramowania. Jest to jeden z procesów zapewnienia jakości oprogramowania. Testowanie ma na celu weryfikację oraz walidację oprogramowania. Weryfikacja oprogramowania pozwala skontrolować, czy wytwarzane oprogramowanie jest zgodne ze specyfikacją. Walidacja sprawdza, czy oprogramowanie jest zgodne z oczekiwaniami użytkownika. Testowanie oprogramowania może być wdrożone w dowolnym momencie wytwarzania oprogramowania (w zależności od stosowanej metody).
- W podejściu kaskadowym zgodnym z modelem V wysiłek zespołu testerskiego zaczyna się wraz z definicją wymagań i jest kontynuowany po zaimplementowaniu zdefiniowanych wymagań.
- Nowsze metody wytwarzania oprogramowania (np. Agile) rozkładają wysiłek testerski równomiernie na poszczególne iteracje i skupiają się na testach jednostkowych oraz automatyzacji weryfikacji wykonywanych przez członków zespołu.

# Testowanie oprogramowania i TDD



## Historia BUGa

9/9


0800 Antan started  
1000 " stopped - antan ✓

1300 (032) MP-MC { 1.2700 9.037 847 025  
                          ~~1.30476415~~ 9.037 846 795 correct  
                          033) PRO 2 2.130476415 4.615925059(-2)  
                          correct 2.130676415

Relays 6-2 in 033 failed special speed test  
in relay 11.000 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
1525 Started Multi-Adder Test.

1545  Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

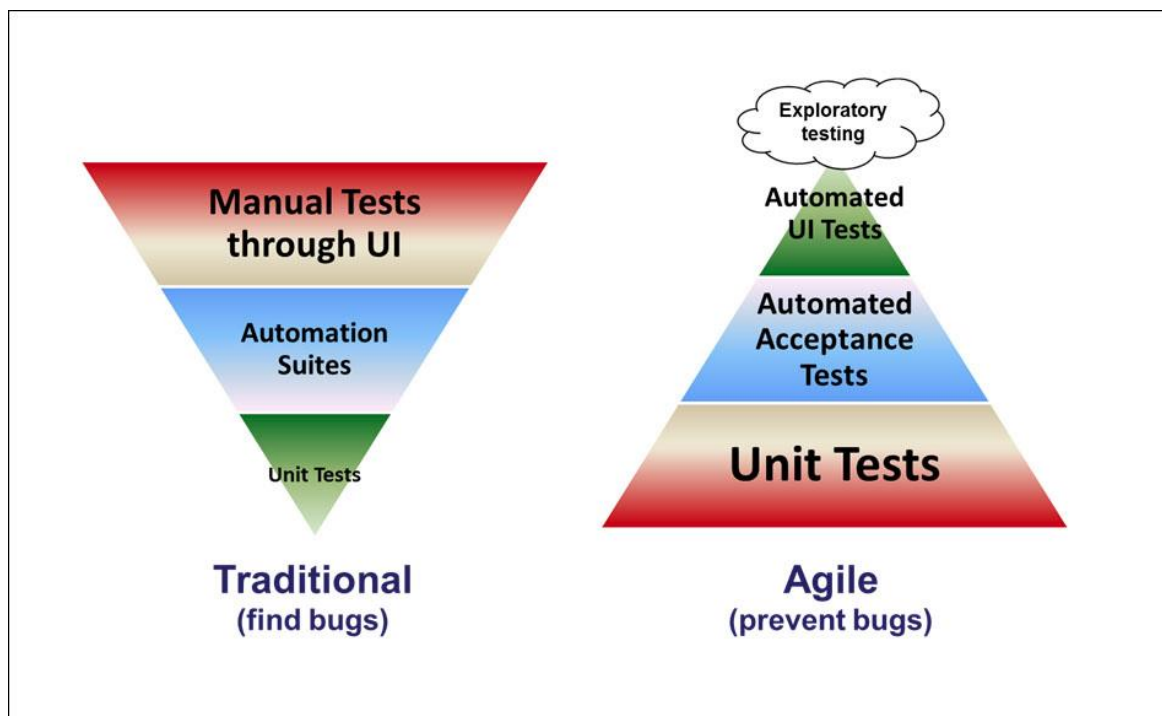
1630 Antan started.  
1700 closed down.

Źródło: [https://pl.wikipedia.org/wiki/B%C5%82%C4%85d\\_\(informatyka\)](https://pl.wikipedia.org/wiki/B%C5%82%C4%85d_(informatyka))

# Testowanie oprogramowania i TDD



## Piramida testów



Źródło: <https://www.agilecoachjournal.com/2014-01-28/the-agile-testing-pyramid>

# Testowanie oprogramowania i TDD

---

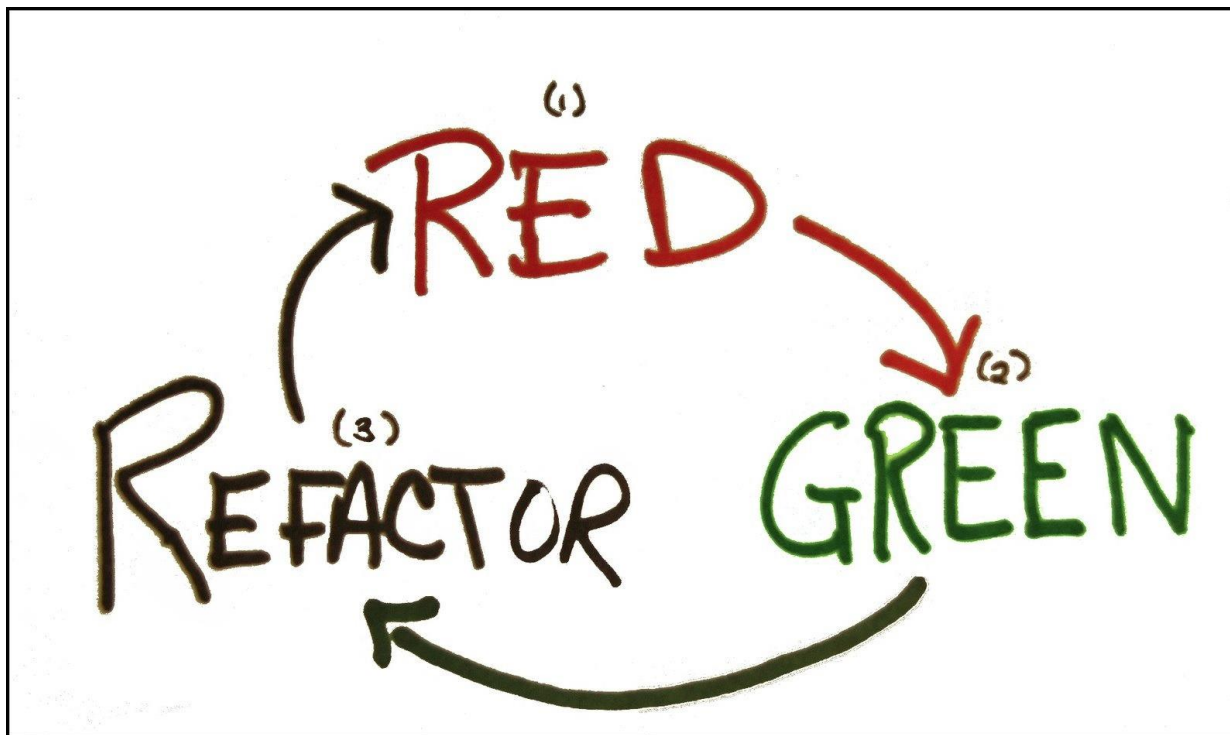


- **Test-driven development (TDD)** – technika tworzenia oprogramowania, zaliczana do metodyk zwinnych. Pierwotnie była częścią programowania ekstremalnego (*extreme programming*), lecz obecnie stanowi samodzielną technikę. Polega na wielokrotnym powtarzaniu kilku kroków:
- Najpierw programista pisze automatyczny test sprawdzający dodawaną funkcjonalność. Test w tym momencie nie powinien się udać.
- Później następuje implementacja funkcjonalności. W tym momencie wcześniej napisany test powinien się udać.
- W ostatnim kroku programista dokonuje refaktoryzacji napisanego kodu, żeby spełniał on oczekiwane standardy.
- Technika została stworzona przez Kenta Becka. Można jej też używać do poprawiania istniejącego kodu.
- Programowanie techniką test-driven development wyróżnia się tym, że najpierw programista zaczyna od pisania testów do funkcji, która jeszcze nie została napisana.

# Testowanie oprogramowania i TDD



## Rytm TDD

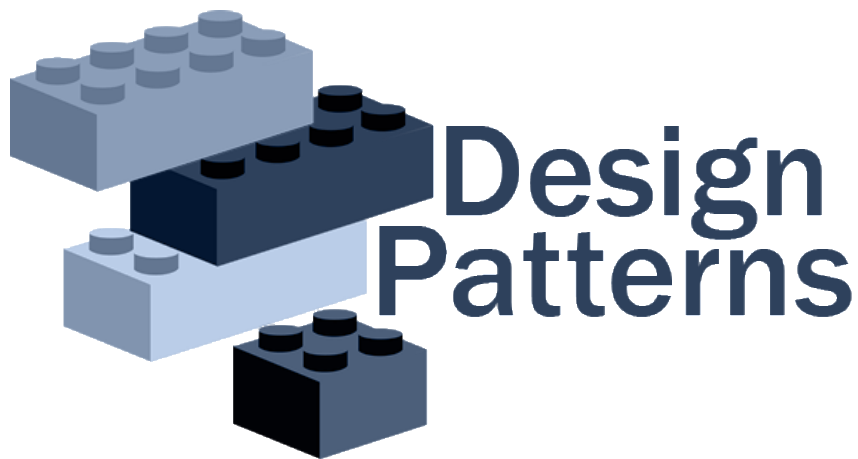


Źródło: <https://airbrake.io/blog/sdlc/test-driven-development>

# Wzorce projektowe



- **Wzorzec projektowy** (*design pattern*) – uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Jest opisem rozwiązania, a nie jego implementacją. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.

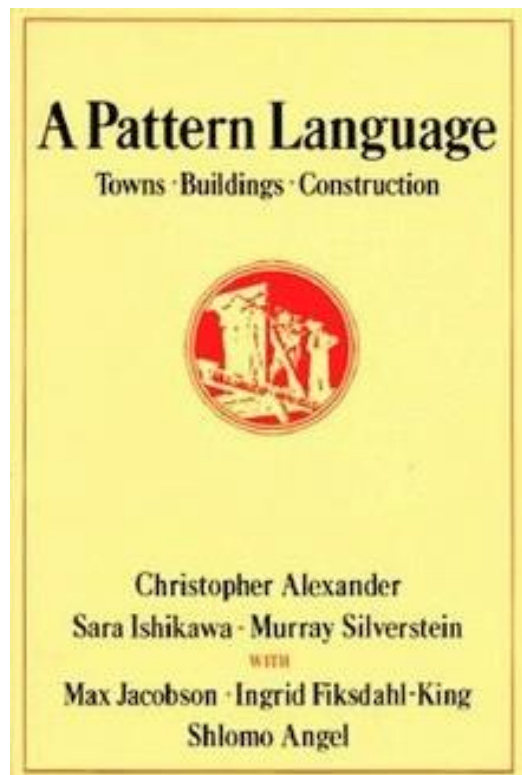


Źródło: <https://bgasparotto.com/design-patterns>





## Język wzorców – Christopher Alexander



Źródło: [https://en.wikipedia.org/wiki/A\\_Pattern\\_Language](https://en.wikipedia.org/wiki/A_Pattern_Language)

# Wzorce projektowe



## GoF – Gang of Four

- GoF's contributions in design pattern
  - Gang of Four was a team of four members: Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.



(L-R) Ralph, Erich, Richard and John

Źródło: <https://www.quora.com/Where-did-design-patterns-come-from>



Źródło: <https://helion.pl/wzoelv.htm#format/b>

# Bazy danych SQL

---

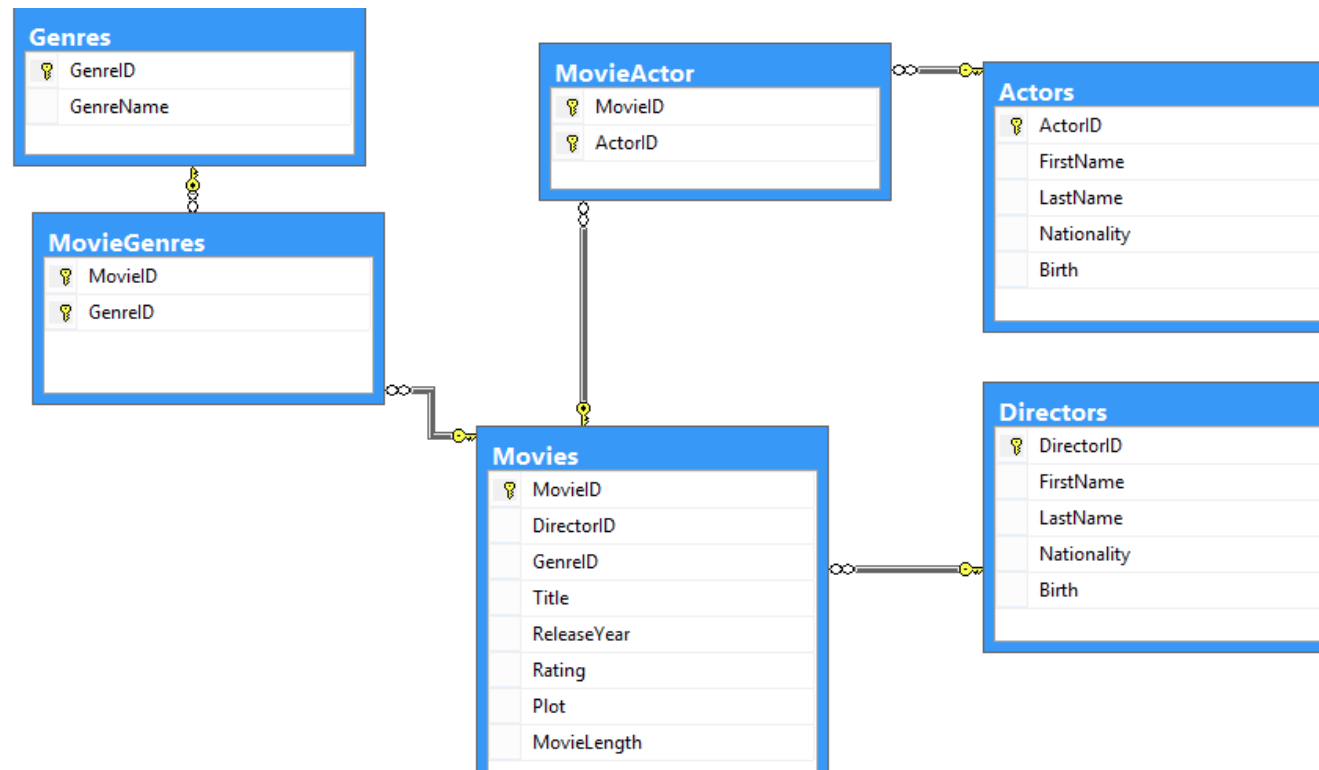


- **Model relacyjny** – model organizacji danych bazujący na matematycznej teorii mnogości, w szczególności na pojęciu relacji. Na modelu relacyjnym oparta jest relacyjna baza danych (*Relational Database*) – baza danych, w której dane są przedstawione w postaci relacyjnej.
- W najprostszym ujęciu w modelu relacyjnym dane grupowane są w relacje, które reprezentowane są przez tablice.
- Relacje są pewnym zbiorem rekordów o identycznej strukturze wewnętrznie powiązanych za pomocą związków zachodzących pomiędzy danymi.
- Relacje zgrupowane są w tzw. schematy bazy danych.
- Relacją może być tabela zawierająca dane teleadresowe pracowników, zaś schemat może zawierać wszystkie dane dotyczące firmy.

# Bazy danych SQL



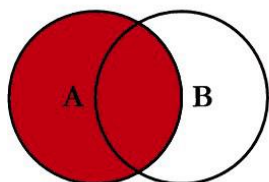
## Relacyjne bazy danych SQL - schemat



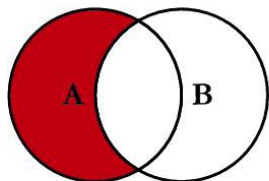


## Złączenia danych

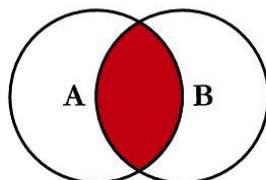
### SQL JOINS



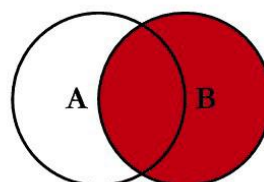
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



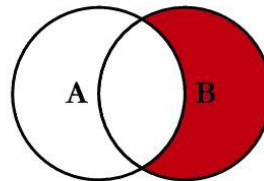
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



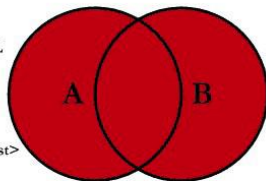
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



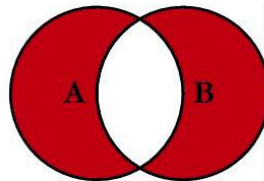
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

© C.L. Moffatt, 2003

Źródło: <https://imgur.com/lrL56lT>

# Bazy danych SQL



- **ACID** – zbiór właściwości gwarantujących poprawne przetwarzanie transakcji w bazach danych. ACID jest skrótem od angielskich słów *atomicity*, *consistency*, *isolation*, *durability*, czyli *niepodzielność*, *spójność*, *izolacja*, *trwałość*.
  - **Niepodzielność** transakcji oznacza, że każda transakcja albo zostanie wykonana w całości, albo w ogóle. Na przykład jeśli w ramach jednej transakcji odbywać się ma przelew bankowy, to nie może dojść do sytuacji, że z jednego konta ubędzie pieniędzy, a kwota na koncie docelowym będzie bez zmian.
  - **Spójność** transakcji oznacza, że po wykonaniu transakcji system będzie spójny, czyli nie zostaną naruszone zasady integralności.
  - **Izolacja** transakcji oznacza, że jeśli dwie transakcje wykonują się współbieżnie, to zwykle nie widzą wprowadzanych przez siebie zmian.
  - **Trwałość** danych oznacza, że system potrafi uruchomić się i udostępnić spójne, nienaruszone i aktualne dane zapisane w ramach zatwierdzonych transakcji, na przykład po nagłej awarii zasilania.



## ACID



Źródło: <https://premaseem.wordpress.com/2016/04/15/what-are-acid-properties-of-database/>

# Bazy danych NoSQL

---



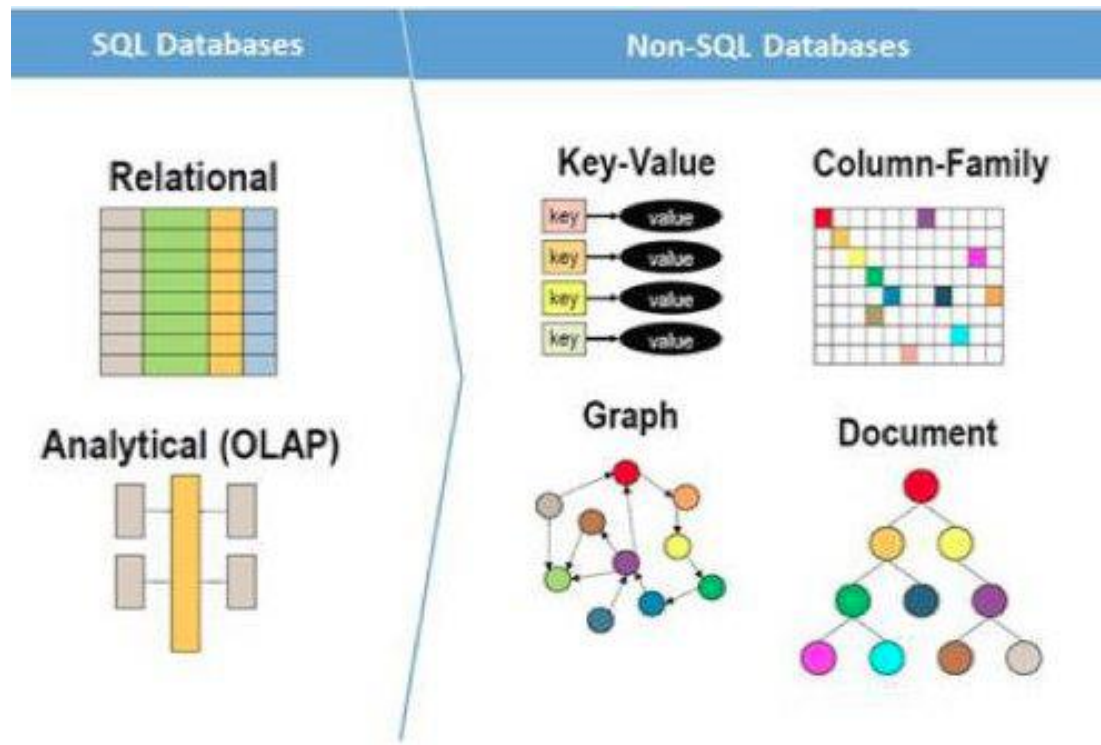
- **NoSQL** (nierelacyjna baza danych SQL, pierwotnie „non SQL” lub „non relational”) – baza danych zapewniająca mechanizm do przechowywania i wyszukiwania danych modelowanych w inny sposób niż relacje tabelaryczne używane w relacjach baz danych SQL.
- Takie bazy danych istniały od końca lat 60. XX wieku, ale nie używano nazwy „NoSQL”.
- NoSQL umożliwia tworzenie prostych projektów, horyzontalne skalowanie do klastrów maszyny (co jest problemem dla relacyjnych baz danych) i lepszą kontrolę nad dostępnością.
- Bazy NoSQL wykorzystywane są coraz częściej w big data działających w czasie rzeczywistym.
- Inne cechy wspólne baz NoSQL to open source, brak schematu danych oraz możliwość wyboru sposobu przechowywania danych w zależności od ich specyfiki.



# Bazy danych NoSQL



## SQL vs NoSQL

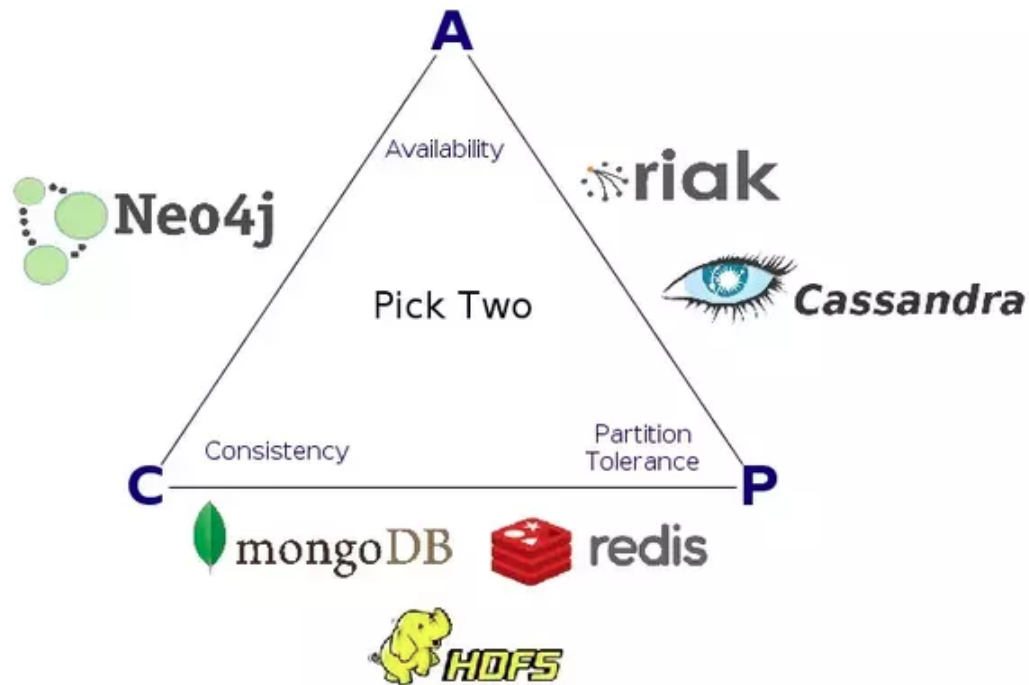


Źródło: <https://medium.com/@pepekamel/sql-vs-nosql-optimal-uses-b722475ec704>

# Bazy danych NoSQL



## CAP theorem



Źródło: <https://www.quora.com/How-many-nodes-are-required-to-set-up-a-distributed-system-in-order-to-test-CAP-theorem>

# JDBC i Hibernate

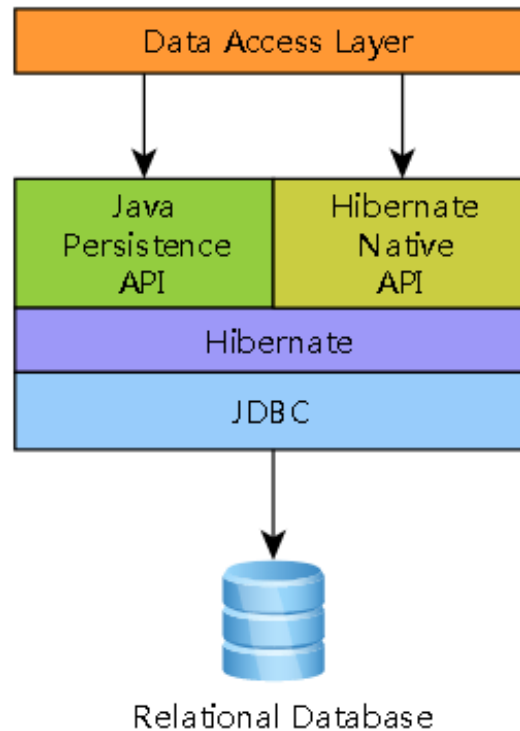


- **JDBC** (*Java DataBase Connectivity*)– interfejs programowania opracowany w 1996 r. przez Sun Microsystems, umożliwiający niezależnym od platformy aplikacjom napisanym w języku Java porozumiewanie się z bazami danych za pomocą języka SQL. Interfejs ten jest odpowiednikiem standardu ODBC opracowanego przez SQL Access Group.
- Środowisko Java zawiera API JDBC, natomiast użytkownik musi uzyskać specjalny sterownik JDBC do swojej bazy danych.
- **Hibernate** - framework do realizacji warstwy dostępu do danych (*persistence layer*). Zapewnia przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiekowym (*O/R mapping*). Dzięki opisowi struktury danych za pomocą języka XML lub adnotacji, umożliwia "rzutowanie" obiektów bezpośrednio na istniejące tabele bazy danych.
- Dodatkowo Hibernate zwiększa wydajność operacji na bazie danych dzięki buforowaniu i minimalizacji liczby przesyłanych zapytań.

# Bazy danych NoSQL



## JDBC vs Hibernate



Źródło: [https://docs.jboss.org/hibernate/stable/orm/userguide/html\\_single/Hibernate\\_User\\_Guide.html](https://docs.jboss.org/hibernate/stable/orm/userguide/html_single/Hibernate_User_Guide.html)

# Spring

---



- **Spring Framework** – szkielet tworzenia aplikacji (*application framework*) w języku Java dla platformy Java Platform, Enterprise Edition.
- Spring Framework powstał na bazie kodu opublikowanego w książce Roda Johnsona Design and Development. Pozytywny odzew czytelników sprawił, że był on dalej rozwijany przez autorów.
- Spring Framework powstał jako alternatywa dla programowania aplikacji z użyciem Enterprise JavaBeans. Programowanie z użyciem EJB narzucało wiele ograniczeń. Funkcjonalność EJB okazała się także za "ciężka" do wszystkich zastosowań; jednocześnie stworzenie małej aplikacji w środowisku EJB wymagało nakładu pracy jak przy aplikacji dużej.
- Odmienna koncepcja Springa – lekkiego szablonu, który nie wymusza specyficznego modelu programowania, stała się bardzo popularna wśród programistów Javy.
- Spring Framework oferuje dużą swobodę w tworzeniu rozwiązań, a jednocześnie jest dobrze udokumentowany i zawiera rozwiązania wielu zagadnień, często występujących w programowaniu.

# Spring

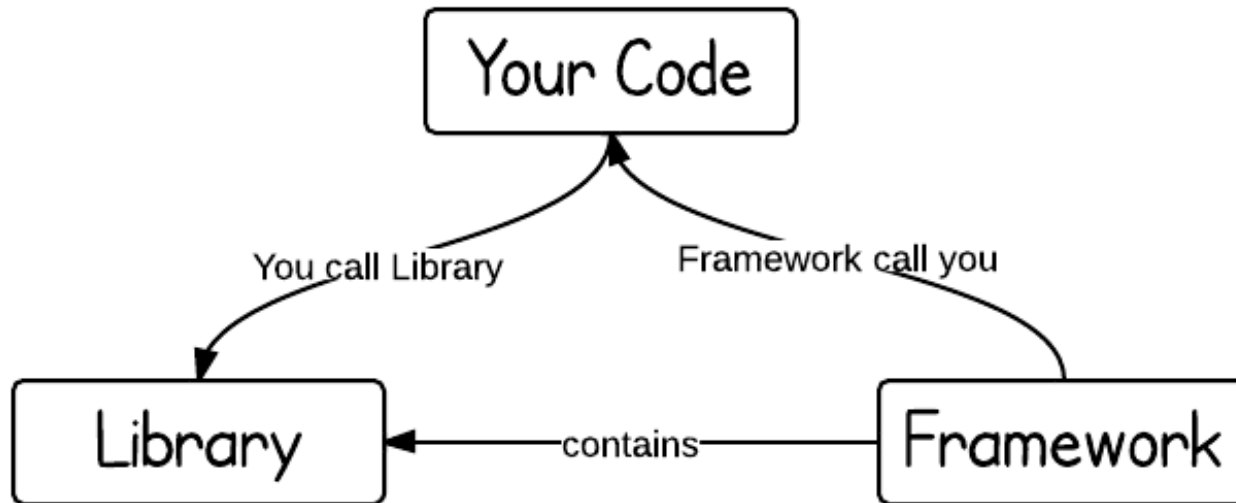
---



- **Biblioteka programistyczna** – plik dostarczający podprogramy, dane oraz typy danych które mogą zostać wykorzystane z poziomu kodu źródłowego programu.
- Użycie bibliotek to sposób na ponowne wykorzystanie tego samego kodu.
- Dobór odpowiedniej biblioteki programistycznej może ułatwić zaimplementowanie niemalże dowolnego zadania.
- Pisanie programów jedynie za pomocą czystych elementów języka programowania jest czasochłonne, a czasami nawet niemożliwe.



## Spring – framework vs biblioteka



Źródło: <https://www.programcreek.com/2011/09/what-is-the-difference-between-a-java-library-and-a-framework/>



- **Framework** – szkielet do budowy aplikacji. Definiuje on strukturę aplikacji oraz ogólny mechanizm jej działania, a także dostarcza zestaw komponentów i bibliotek ogólnego przeznaczenia do wykonywania określonych zadań. Programista tworzy aplikację, rozbudowując i dostosowując poszczególne komponenty do wymagań realizowanego projektu, tworząc w ten sposób gotową aplikację.
- Frameworki bywają niekiedy błędnie zaliczane do bibliotek programistycznych. Typowe cechy, które każą wyróżniać je jako samodzielną kategorię oprogramowania, to:
  - **odwrócenie sterowania** – w odróżnieniu od aplikacji oraz bibliotek, przepływ sterowania jest narzucany przez framework, a nie przez użytkownika.
  - **domyślne zachowanie** – domyślna konfiguracja frameworka musi być użyteczna i dawać sensowny wynik, zamiast być zbiorem pustych operacji do nadpisania przez programistę.
  - **rozszerzalność** – poszczególne komponenty frameworka powinny być rozszerzalne przez programistę, jeśli ten chce rozbudować je o niezbędne mu dodatkowe funkcje.
  - **zamknięta struktura wewnętrzna** – programista może rozbudowywać framework, ale nie poprzez modyfikację domyślnego kodu.



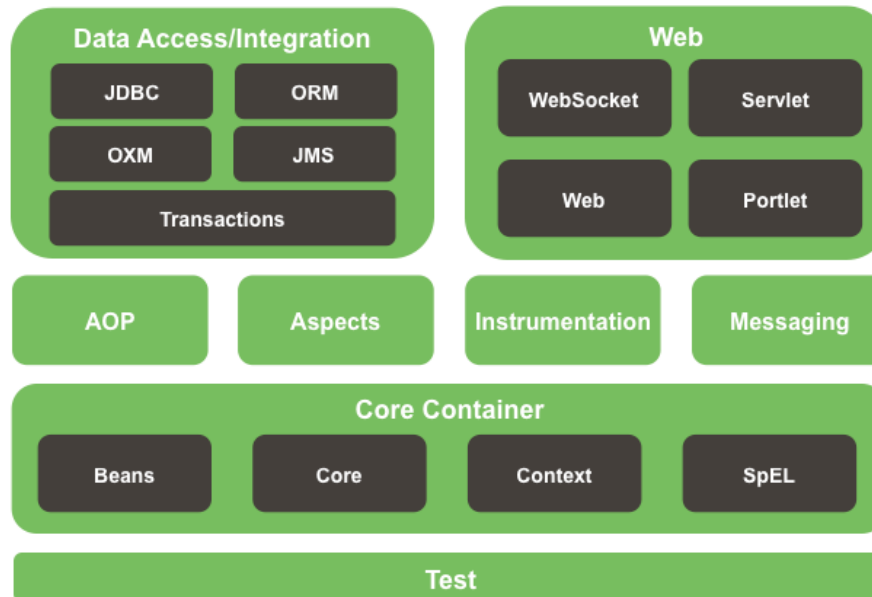


## Spring

-



### Spring Framework Runtime



Źródło: <https://docs.spring.io/spring/docs/4.3.18.RELEASE/spring-framework-reference/htmlsingle/>

# Agile

---



- **Programowanie zwinne** (*agile software development*) – grupa metod wytwarzania oprogramowania opartego na programowaniu iteracyjno-przyrostowym, powstałe jako alternatywa do tradycyjnych metod typu waterfall.
- Najważniejszym założeniem metodyk zwinnych jest obserwacja, że wymagania odbiorcy (klienta) często ewoluują podczas trwania projektu.
- Oprogramowanie wytwarzane jest przy współpracy samzarządzalnych zespołów, których celem jest przeprowadzanie procesów wytwarzania oprogramowania.
- Pojęcie zwinnego programowania zostało zaproponowane w 2001 w **Agile Manifesto**.



## Manifest Agile

### **Manifest programowania zwinnego**

Odkrywamy nowe metody programowania dzięki praktyce w programowaniu i wspieraniu w nim innych.

W wyniku naszej pracy, zaczęliśmy bardziej cenić:

**Ludzi i interakcje** od procesów i narzędzi

**Działające oprogramowanie** od szczegółowej dokumentacji

**Współpracę z klientem** od negocjacji umów

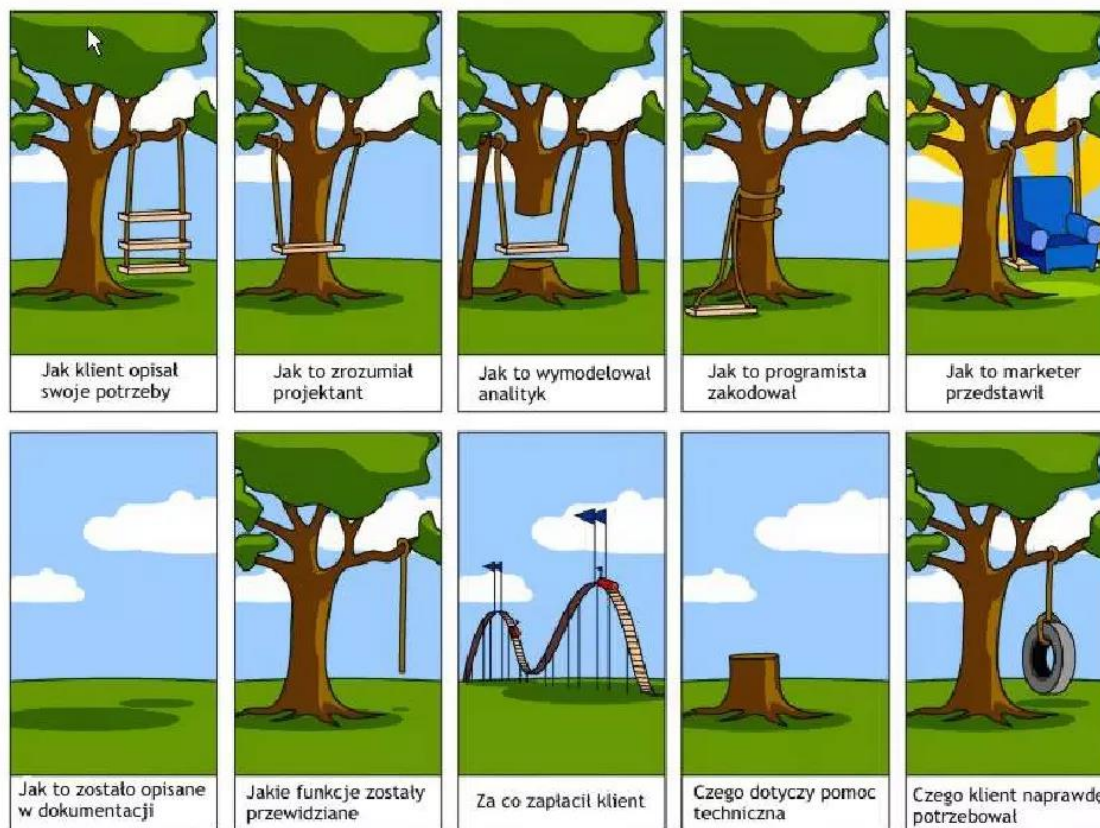
**Reagowanie na zmiany** od realizacji założonego planu.

Oznacza to, że elementy wypisane po prawej są wartościowe, ale większą wartość mają dla nas te, które wypisano po lewej.

Źródło: <http://agilemanifesto.org/iso/pl/manifesto.html>



## Problemy w projektach



Źródło: <http://www.arturnet.pl/scrum-model-kaskadowy/>

# Agile

---



Przy realizacji zadań w projektach w oparciu o metodologię programowania zwinnego "Agile" wyróżnia się następujące po sobie etapy:

- **Plan** (Planowanie)
- **Design** (Projektowanie)
- **Develop** (Programowanie)
- **Test** (Testowanie)
- **Release** (Implementacja)
- **Feedback** (Informacja zwrotna)

# Agile

---



- **Planowanie** - zbieranie dokładnych wymagań od klienta dotyczących danego zadania, ich analiza oraz zaplanowanie kroków koniecznych do realizacji danego zadania w oparciu o pozyskane informacje.
- To bardzo ważny etap mający kluczowe znaczenie dla czasu i jakości realizowanego zadania dlatego nigdy nie powinien być pomijany (jak zresztą wszystkie pozostałe etapy).
- Etap ten wymaga bezpośredniego kontaktu z klientem, a co za tym idzie umiejętności dobrej komunikacji i zrozumienia drugiej osoby.
- Etap ten jest czasem pomijany przez osoby i zespoły nie posiadające umiejętności komunikacji, odczuwające niechęć do komunikacji lub obawiające się komunikacji z klientem.
- Takie osoby lub zespoły nie powinny korzystać z metodologii Agile lecz skorzystać z bardziej sformalizowanych metod i/lub z dedykowanych do tego etapu analityków, którzy będą odpowiedzialni za dokładne zebranie wymagań i ich analizę.

# Agile



- **Projektowanie** - projektowanie wykonania danego elementu będącego celem zadania na bazie informacji zebranych na etapie Planowania.
- Można to porównać do wykonania projektu elementu domu przez architekta przed przystąpieniem do prac nad nim przez ekipę budowlaną.
- Ten etap jest również czasem pomijany przez osoby lub zespoły, które mylnie rozumieją fazę projektowania jako wykonywanie dokumentacji powykonawczej uznając ją w związku z tym za zbędną.
- **Programowanie** - właściwy etap prac nad danym zadaniem na bazie przygotowanego projektu zadania
- **Testowanie** - testowanie danego elementu będącego podmiotem zadania od strony technicznej przez osobę lub zespół wykonujący dane zadanie oraz od strony klienta (User Acceptance Test) czy dany element jest tym czego klient oczekiwał

# Agile

---



- **Implementacja** - po pozytywnych wynikach testów zarówno technicznych jak i klienckich (akceptacji przez klienta) przekazanie danego elementu projektu na "produkcję" do finalnego użytkowania przez klienta
- **Informacja zwrotna** - przekazanie informacji zwrotnej przez klienta do osoby lub zespołu realizującego dane zadanie w projekcie odnośnie ewentualnych mniejszych błędów, których nie wykryto podczas testów, zgłaszanie potencjalnych usprawnień do realizacji w kolejnych cyklach lub zgłoszenie zmiany wymagań klienta.
- Mając na myśli zmianę wymagań mówimy o rzeczywistej zmianie wymagań - wynikłej np. ze zmiany procesów lub potrzeb klienta - nie natomiast jak zaznaczono wcześniej o zmianie wynikającej z pominięcia lub niedokładnego przeprowadzenia etapu Planowania.



# Agile

---



- **Scrum** – iteracyjne i przyrostowe ramy postępowania zgodne ze Scrum Guide. Może mieć zastosowanie w realizacji projektów w oparciu o metodyki zwinne zgodne z manifestem Agile.
- Rozwój produktu podzielony jest na trwające maksymalnie jeden miesiąc iteracje, zwane sprintami.
- Po każdym sprincie zespół powinien być w stanie dostarczyć działającą wersję produktu.
- Scrum jest często stosowany podczas tworzenia i rozwijania oprogramowania, nie jest jednak ograniczony tylko do tej dziedziny.

# Agile

---



- Zespół pracuje w określonych przedziałach czasowych zwanych przebiegami lub sprintami (*sprint*).
- Zmiany wprowadzane w każdym przebiegu powinny wносить zauważalną dla użytkowników nową wartość funkcjonalną.
- Przebieg nie może trwać dłużej niż jeden miesiąc. W praktyce sprinty trwają od 1 do 4 tygodni.
- Zaleca się stosowanie przebiegów o stałych długościach.
- Na początku pracy nad produktem zbierana jest lista wymagań użytkownika, są one przeważnie gromadzone w postaci historyjek (*User Stories*).
- Każda historyjka opisuje jedną cechę systemu.
- Właściciel produktu (*Product Owner*) jest też zobowiązany do przedstawienia priorytetu wymagań oraz głównego celu pierwszego przebiegu.
- Po tym formułowany jest rejestr wymagań (*Product Backlog*).
- Cel przebiegu jest zapisywany w widocznym miejscu w pokoju członków zespołu.

# Agile

---



- Następnie podczas planowania przebiegu (*Sprint Planning*) wybierane są zadania o najwyższym priorytecie, a jednocześnie przyczyniające się do realizacji celu przebiegu.
- Szacuje się czas realizacji, pracochołność, złożoność i ryzyko każdego zadania. Lista zadań wraz z oszacowaną czasochłonością nosi nazwę rejestru zadań przebiegu (*Sprint Backlog*).
- Po planowaniu zespół przechodzi do realizacji przebiegu. W jego trakcie Właściciel Produktu powinien cały czas pracować z zespołem nad jak najlepszym zrozumieniem wymagań nie ingerując jednocześnie w sposób ich implementacji.
- Nie powinno się także zmieniać zakresu Sprintu.
- Zespół z założenia jest ciałem samoorganizującym się, nie ma mowy o odgórnym przypisywaniu zadań poszczególnym członkom zespołu. Dokonują oni wyboru realizowanych zadań, według wspólnych ustaleń, umiejętności czy innych preferencji.

# Agile

---



- Naczelną zasadą metody jest przeprowadzanie codziennych (maksymalnie 15-minutowych) spotkań (*Daily Scrum*), na których omawiane są zadania zrealizowane poprzedniego dnia, problemy występujące przy ich realizacji oraz zadania do wykonania w dniu spotkania.
- Każdy Sprint kończy się spotkaniem (*Sprint Review*), na którym prezentowany jest produkt wykonany podczas przebiegu. Powinni w nim uczestniczyć wszyscy zainteresowani projektem.
- Na spotkaniu każdy członek zespołu może zabrać głos i wyrazić opinię o produkcie.
- Na zakończenie ustalany jest termin spotkania planistycznego do następnego przebiegu.
- Realizacja projektu według Scrum skupia się na:
  - dostarczaniu kolejnych, coraz bardziej dopracowanych wyników projektu,
  - włączaniu się przyszłych użytkowników w proces wytwórczy,
  - samoorganizacji zespołu projektowego.

# Agile

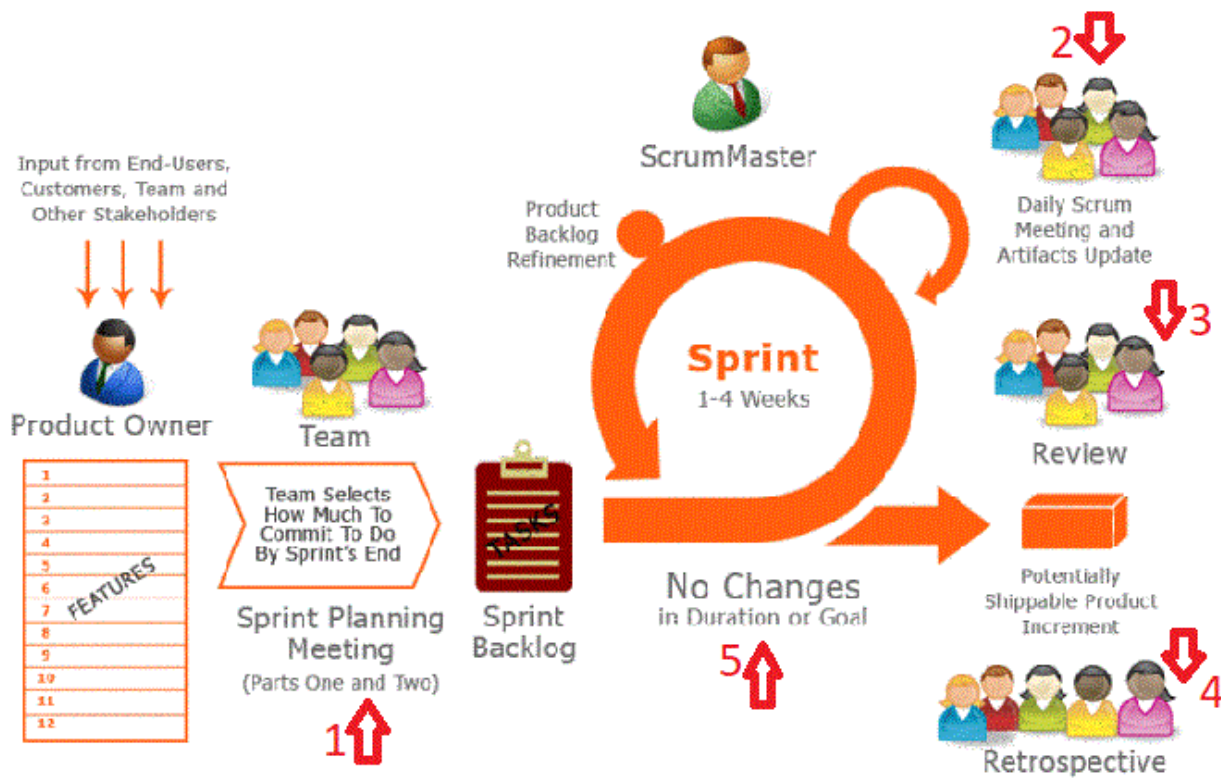
---



- Zazwyczaj Zespół Scrum składa się z od 3 do 9 osób. Dobrze, gdy ma charakter interdyscyplinarny i składa się z osób reprezentujących różne umiejętności. Główne role w projekcie grają: Scrum Master, Właściciel Produktu (*Product Owner*) i Zespół Deweloperski (*Development Team*).
- **Zespół Deweloperski** – grupa osób, składająca się z od trzech do dziewięciu osób, odpowiedzialna za dostarczenie produktu
- **Właściciel produktu** – osoba reprezentująca klienta. Właściciel produktu może być członkiem zespołu, jednak nie jest zalecane, aby jednocześnie był Scrum Masterem.
- **Scrum Master** – osoba odpowiedzialna za usuwanie wszelkich przeszkód uniemożliwiających zespołowi wykonanie zadania, oraz za poprawną implementację procesu i metod.



## Scrum



Źródło: <https://www.c-sharpcorner.com/article/scrum-framework-5-events-in-scrum-framework/>



# DZIĘKUJĘ ZA UWAGĘ



WYKONAWCASZKOLENIA:

software  
development  
academy



Szkolenia  
Komputerowe



## BIURO PROJEKTU

Toruńska Agencja Rozwoju Regionalnego S.A.

ul. Włocławska 167, 87-100 Toruń

56 699 54 89

szkoleniakomputerowe@tarr.org.pl

www.szkoleniakomputerowe.tarr.org.pl

Projekt „CERTYFIKOWANE SZKOLENIA KOMPUTEROWE dla osób dorosłych z województwa kujawsko-pomorskiego”  
współfinansowany ze środków Europejskiego Funduszu Społecznego w ramach Regionalnego Programu Operacyjnego  
Województwa Kujawsko-Pomorskiego na lata 2014 – 2020