

Prawa autorskie dla programisty – wersja skrócona na zajęcia

Zawiera:

- Prawa majątkowe
- Rodzaje licencji
- Zawartość pliku License





Czym jest prawo autorskie?

- Prawo chroniące twórców
- Chroni sposób wyrażenia, a nie pomysł
- Działa automatycznie

```

if __name__ == "__main__":
    loadMatlabData()
    MT = MultiTankModel(interp1dX = dcPumpX,
                        interp1dY = dcPumpY,
                        interp2dX = H,
                        interp2dY = PWM,
                        interp2dZ_t1 = ev1,
                        interp2dZ_t2 = ev2,
                        interp2dZ_t3 = ev3)

    start = time.clock()
    with open('./simLog.csv','w') as f:
        f.write('')

    simulate = True
    plotting = True
    dt = np.float64(0.2)

    simTime = 10000
    prevFileChangedTime = 0
    currFileChangedTime = 0
    fileName = 'C:\OMRON\Data\SimulatorData\CARD\Memory001\simRWFile'

    while simulate and MT.time<=simTime:
        try:
            currFileChangedTime = os.stat(fileName).st_mtime
            if prevFileChangedTime != currFileChangedTime:
                exchangeFileData(fileName)
                currFileChangedTime = os.stat(fileName).st_mtime
                MT.simulate(samplingTime = dt)
                appendLogData()
                print('simulation time:',MT.time,' with step:',MT.step, \
                    "t1.h=%s, t2.h=%s, t3.h=%s" %(MT.tank1.height*100,MT.tank2.

            prevFileChangedTime = currFileChangedTime
            time.sleep(0.005)
        except KeyboardInterrupt:
            print("simulation paused, type 'exit' to stop or type float value o
            stringInput = input("command = ")
            if stringInput == "exit":
                simulate = False
            else:
                try:
                    dt = float(stringInput)
                    print("simulation time delta changed, dt=%s" %dt)
                except:
                    print("wrong command")

    print("total script time:",time.clock()-start)
    if plotting:
        plotAll()

```

Kod jako utwór

- Program komputerowy to utwór
- Musi być twórczy i oryginalny
- Nie musi być duży ani skończony

Przykład – kod jako utwór



APLIKACJA W
PYTHONIE



KALKULATOR,
GRA, FORMULARZ



PROJEKT TKINTER
NA LEKCJI

Kto jest autorem kodu?

Zawsze osoba
fizyczna

Uczeń = autor
projektu

AI i firma nie
są autorami

Prawa osobiste autora



AUTORSTWO



PODPISANIE KODU



OCHRONA PRZED
ZMIANAMI

Prawa majątkowe autora



Umowa a prawa autorskie



UMOWA DECYDUJE
O PRAWACH



BEZ UMOWY
PRAWA MA AUTOR



DOTYCZY TEŻ
UCZNIÓW

Przykład – projekt na zlecenie



STRONA WWW DLA
KLIENTA



BEZ UMOWY – PRAWA
U PROGRAMISTY



Z UMOWĄ – PRAWA U
KLIENTA

Open Source – co to jest?



Darmowe
oprogramowanie



Z określonymi
zasadami



Każda licencja ma
warunki

Popularne licencje OSS



CZYM JEST OPEN SOURCE?



**OPEN SOURCE
SOFTWARE TO
OPROGRAMOWANIE,
KTÓREGO:**



**KOD ŹRÓDŁOWY
JEST PUBLICZNIE
DOSTĘPNY,**



**MOŻNA GO UŻYWAĆ,
MODYFIKOWAĆ I
ROZPOWSZECHNIĄĆ,**

Popularne licencje OSS

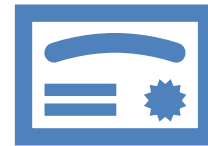
... ALE zawsze na warunkach licencji.

Open Source ≠ „rób co chcesz”

Dlaczego licencja jest ważna?

Licencja określa:

- czy wolno używać kodu komercyjnie,
- czy wolno go modyfikować,
- czy trzeba podać autora,
- czy trzeba udostępnić własny kod.



Popularne licencje OSS

Licencje „luźne” (permissive):

MIT License

Najpopularniejsza licencja OSS.

Można:

- używać w projektach szkolnych,
- używać komercyjnie,
- modyfikować kod,
- zamknąć kod (closed source).

Warunek:

- trzeba zachować informację o autorze i licencji.

Przykład zastosowania:

Uczeń używa biblioteki MIT w aplikacji Java — **legalne**.

Apache License 2.0

Bardziej „rozbudowana”.

Można:

- używać komercyjnie,
- modyfikować,
- łączyć z innym kodem.

Dodatkowo:

ochrona przed roszczeniami patentowymi.

Warunek:

- zachować licencję,
- zaznaczyć zmiany w kodzie.

Przykład zastosowania:

Częsta w projektach Google.

BSD License

Podobna do MIT.

Można:

wszystko jak w MIT.

Warunek:

- zachowanie informacji o autorze.

Różnica:

- często spotykana w systemach operacyjnych.

Przykład zastosowania:

- w systemach operacyjnych,
- w sieciach,
- w embedded.

Popularne licencje OSS

Licencje „wirusowe” (copyleft):

GPL (General Public License)

Licencja „z zasadą dziedziczenia”.

Jeśli:

- użyjesz kodu GPL,
- i stworzysz projekt pochodny,

Twój projekt MUSI być też GPL.

Można:

- używać,
- modyfikować,
- rozpowszechniać.

Obowiązek:

- udostępnić kod źródłowy.

Przykład zastosowania:

Nie można użyć biblioteki GPL w zamkniętej aplikacji komercyjnej.

LGPL (Lesser GPL)

Łagodniejsza wersja GPL.

Można:

- używać biblioteki LGPL w projekcie zamkniętym,
- jeśli tylko ją linkujesz.

Nie można:

modyfikować biblioteki bez udostępnienia zmian.

Przykład zastosowania:

Częsta w bibliotekach.

Inne popularne licencje

MPL (Mozilla Public License):

- pośrednia między MIT a GPL.
- zmodyfikowane pliki muszą być open source,
- reszta projektu może być zamknięta.

Przykład zastosowania:

Używana m.in. w Firefoksie.

Którą licencję wybrać?

Chcę:

- **pełną swobodę** → MIT / BSD
- **ochronę patentową** → Apache 2.0
- **zmuszać innych do open source** → GPL
- **kompromis** → LGPL / MPL

Plik LICENSE:

- Określa zasady użycia
- Chroni autora

Licencja BSD 3-Clause (wersja polska)

Copyright (c) 2026 [Twoje imię i nazwisko]
Wszelkie prawa zastrzeżone.

Niniejszym zezwala się na redystrybucję i używanie tego oprogramowania w formie źródłowej i binarnej, z modyfikacjami lub bez, pod warunkiem spełnienia poniższych warunków:

1. Redystrybucja kodu źródłowego musi zawierać powyższą informację o prawach autorskich, niniejszą listę warunków oraz poniższe zastrzeżenie odpowiedzialności.
2. Redystrybucja w formie binarnej musi zawierać powyższą informację o prawach autorskich, niniejszą listę warunków oraz poniższe zastrzeżenie odpowiedzialności w dokumentacji i/lub innych materiałach dostarczanych wraz z oprogramowaniem.

3. Nazwisko autora ani nazwy współtwórców nie mogą być wykorzystywane do promowania produktów opartych na tym oprogramowaniu bez uprzedniej pisemnej zgody autora.

NINIEJSZE OPROGRAMOWANIE JEST DOSTARCZANE „TAK JAK JEST”, BEZ JAKIEJKOLWIEK GWARANCJI, WYRAŹNEJ LUB DOROZUMIANEJ, W TYM MIĘDZY INNYMI GWARANCJI PRZYDATNOŚCI HANDLOWEJ LUB PRZYDATNOŚCI DO OKREŚLONEGO CELU.

AUTOR ANI WSPÓŁTWÓRCY NIE PONOSZĄ ODPOWIEDZIALNOŚCI ZA JAKIEJKOLWIEK SZKODY POWSTAŁE W WYNIKU UŻYWANIA TEGO OPROGRAMOWANIA, W TYM SZKODY BEZPOŚREDNIE, POŚREDNIE, LOSOWE LUB WYNIKOWE.

Położenie pliku w strukturze

```
projekt/  
├─ src/  
│   └─ main.py  
├─ README.md  
└─ LICENSE
```

W folderze projektu obok katalogów z kodem źródłowym. Najczęściej w języku angielskim.

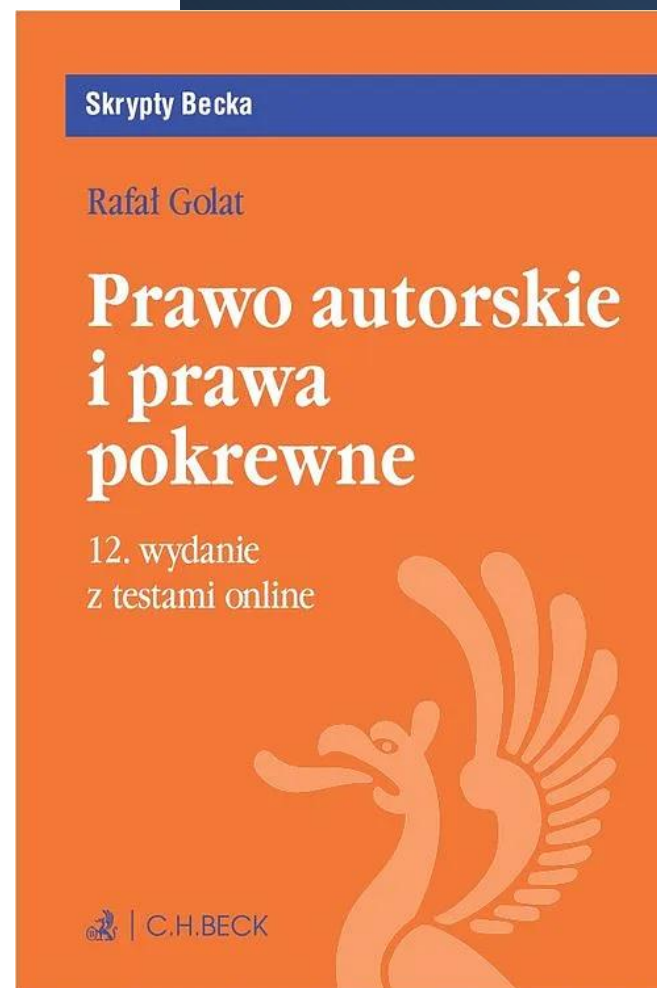


Przykład – projekt ucznia

*Projekt na GitHub +
Dodana licencja MIT
= Autor zachowuje
prawa*

Czego NIE wolno robić!

- Kopiować kodu bez zgody
- Udawać autora
- Ignorować licencji



Najczęstsze błędy uczniów

- Kod z Internetu bez licencji
- Brak autorstwa
- Brak LICENSE



License

[Traditional Licensing](#)

Generate and manage PAK-based and other device licenses, including demo licenses

[Smart Software Licensing](#)

Track and manage Smart Software Licenses.

[Enterprise Agreements](#)

Generate and manage licenses from Enterprise Agreements.

Podsumowanie



KOD TO UTWÓR



AUTOR TO
PROGRAMISTA



LICENCJA I UMOWA
SĄ KLUCZOWE