Programowanie obiektowe

2021L

Informacje organizacyjne

Informacje organizacyjne

Prowadzący

- Jacek Starzyński;
- Paweł Zawadzki.

Informacje organizacyjne

Prowadzący

Plan kursu

- Cykl życia oprogramowania;
- Python – wprowadzenie;
- paradygmat programowania obiektowego;
- wzorce projektowe *;
- aplikacje konsolowe, GUI;
- zaawansowane biblioteki *;
- przetwarzanie równoległe i rozproszone.

* wybrane elementy (nie wszystkie!)

Informacje organizacyjne

Prowadzący

Plan kursu

Zajęcia

- 30 godzin wykładu;
- 30 godzin laboratorium.

Informacje organizacyjne

Prowadzący
Plan kursu
Zajęcia
Zaliczenie

- Zajęcia składowe kursu podlegają łącznemu zaliczeniu;
- z każdej części (wykład, laboratorium) trzeba zdobyć powyżej 50% możliwych punktów (odpowiednio: 51 i 36 punktów);
- na 14. wykładzie: zaliczenie (7 pytań; od 0 pkt. do 10 pkt.);
- na 15. wykładzie: omówienie zaliczenia;
- w przypadkach uzasadnionych: do uzgodnienia dodatkowy termin zaliczenia.

Informacje organizacyjne

Prowadzący
Plan kursu
Zajęcia
Zaliczenie
Ocena końcowa

Ocena końcowa: (liczba punktów z zaliczenia) + (liczba punktów z laboratorium) * 0.3.

Punkty	Ocena
0-50	2.0
51-60	3.0
61-70	3.5
71-80	4.0
81-90	4.5
91-100	5.0

Informacje organizacyjne

Prowadzący
Plan kursu
Zajęcia
Zaliczenie
Ocena końcowa
Literatura

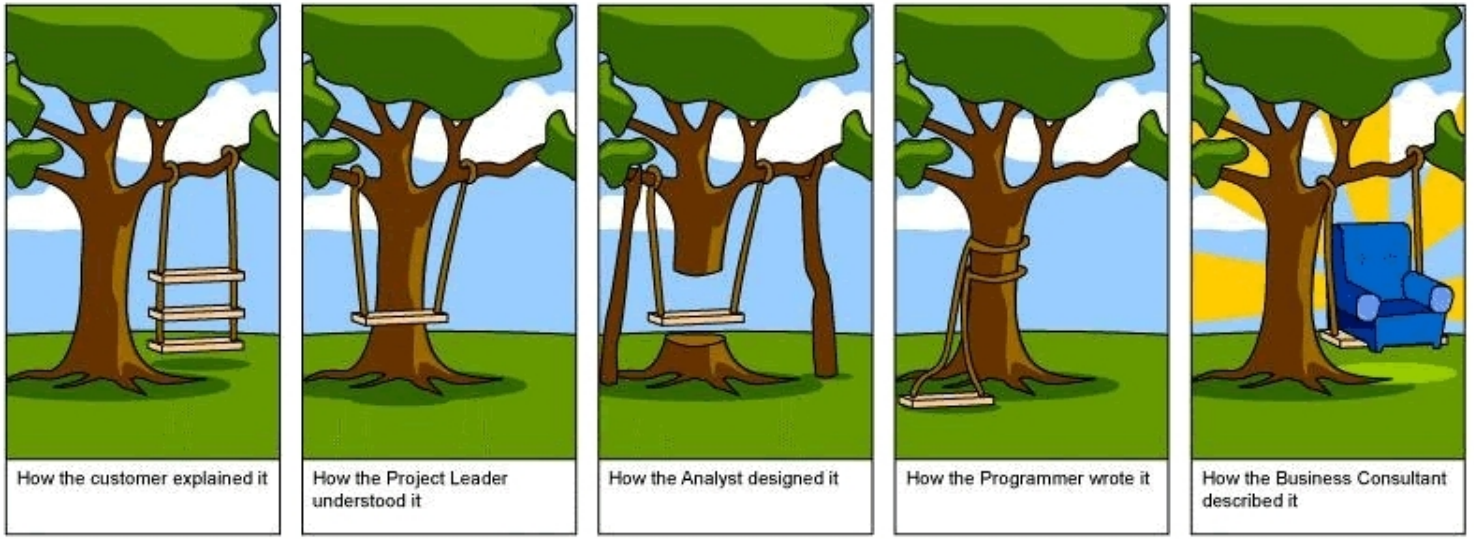
- Mark Lutz, *Python. Wprowadzenie.*, Wydanie IV (?), Helion.

Cykl życia oprogramowania

Cykl życia oprogramowania

Fazy

- Specyfikacja wymagań;
- analiza;
- projektowanie;
- implementacja;
- testowanie (weryfikacja poprawności działania; walidacja z oczekiwaniami klienta);
- wdrożenie;
- utrzymanie / pielęgnacja / rozwój.



https:// www.flickr.com/photos/bandon1/14470581

Cykl życia oprogramowania

Fazy

Modele / metodyki (wybrane)

- Kaskadowa;
- prototypowania;
- spiralna:
  - cel, koncepcja;
  - analiza ryzyka;
  - rozwój oprogramowania (implementacja, testowanie, wdrożenie);
  - weryfikacja oraz planowanie kolejnego etapu lub zakończenie;
- XP (eXtreme Programming);
- Scrum *.

Narysować?

* Metodyka prowadzenia projektu (nie tylko oprogramowania).

// DDD, TDD BDD -- o tym później (jeśli zdążymy)

Cykl życia oprogramowania

Fazy

Modele / metodyki (wybrane)

... Scrum

Role:

- scrum master;
- właściciel produktu (ang. product owner);
- zespół (ang. development team);

Cykl życia oprogramowania

Fazy

Modele / metodyki (wybrane)

... Scrum

Elementy składowe, rytuały:

- sprint;
- historyjki użytkownika (ang. user stories);
- rejestr wymagań / zadań do projektu (ang. product backlog);
- planowanie sprintu (ang. sprint planning);
- rejestr wymagań / zadań do sprintu (ang. sprint backlog);
- codzienne spotkanie (ang. daily scrum);
- spotkanie kończące sprint (ang. sprint review).

Cykl życia oprogramowania

Fazy

Modele / metodyki (wybrane)

... Scrum

Ważne:

- zespół sam przydziela zadania;
- nie zmienia się zakresu sprintu.

Cykl życia oprogramowania

Fazy

Modele / metodyki (wybrane)

... Scrum

Manifest agile

Manifest zwinnego wytwarzania oprogramowania (2001 r.)

- **ludzie i interakcje** ponad procesy i narzędzia;
- **działające oprogramowanie** ponad obszerną dokumentację;
- **współpraca z klientem** ponad formalne ustalenia;
- **reagowanie na zmiany** ponad podążanie za planem.

Python

Python

Filozofia

```
user@host python3
In [1]: import this


The Zen of Python, by Tim Peters
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
...
```

Python

Filozofia

Cechy

- Uniwersalny;
- wieloparadygmatowy:
  - skryptowy;
  - strukturalny;
  - obiektowy;
  - funkcyjny;
- przenośny (byte code);
- czytelny, zwięzły;
- duża społeczność programistów.

Python

Filozofia

Cechy

Historia, rozwój

- Guido van Rossum – autor (90');
- Latający cyrk Monty Pythona;
- proces akceptacji zmian (musi zaakceptować van Rossum).

Python

Filozofia

Cechy

Historia, rozwój

W czym pisać?

- powłoka (*python*);
- interaktywna powłoka (*ipython*);
- pliki źródłowe (*.py*) w dowolnym edytorze tekstowym;
- IDE: PyCharm.

Skończ mówić i pokaż nam kod!

Dziękuję za uwagę.