

Highlight-Save

Created : 2019/10/01

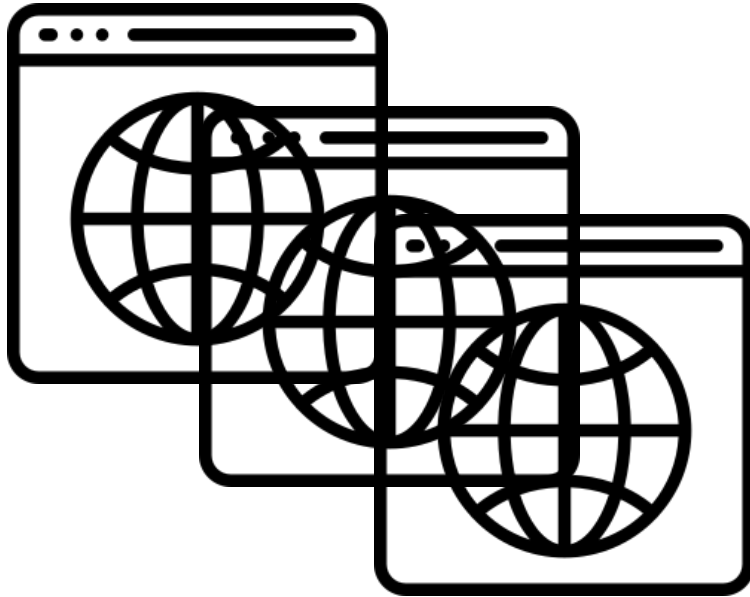
Updated : 2019/10/01

What is it ?

<div>Icons made by Smashicons from www.flaticon.com</div>

<div>Icons made by Smashicons from www.flaticon.com</div>

Pages



Summarize



Documents



How's this will work?

1. Highlight some content in Page
2. Mark each part something like (Title, content, ..)
3. Save that to File (maybe PDF, Word, Text)

1. HighLight

로그인하지 않음 토론 기여 계정 만들기 로그인

위키백과
우리 모두의 백과사전

대문
사용자 모임
요즘 화제
최근 바뀐
모든 문서 보기
임의 문서로
도움말
기부

도구
여기를 가리키는 문서
가리키는 글의 최근 바뀐
파일 올리기
특수 문서 목록
고유 링크
문서 정보
위키데이터 항목
이 문서 인용하기

다른 프로젝트
위키미디어 공용

인쇄/내보내기
책 만들기
PDF로 다운로드
인쇄용 판

다른 언어

العربية
English

읽기

편집

역사 보기

위키백과 검색



Wiki Loves Monuments: 문화재 사진을 찍어 위키백과에 기여하고, 우승에도 도전해 보세요!

더 알아보기



객체 지향 프로그래밍

위키백과, 우리 모두의 백과사전.

객체 지향 프로그래밍(영어: Object-Oriented Programming, OOP)은 **컴퓨터 프로그래밍의 패러다임 중 하나이다.** 객체 지향 프로그래밍은 **컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 "객체"들의 모임으로 파악하고자 하는 것이다.** 각각의 객체는 **메시지**를 주고받고, **데이터**를 처리할 수 있다.

객체 지향 프로그래밍은 **프로그램을 유연하고 변경이 용이하게 만들기 때문에 대규모 소프트웨어 개발에 많이 사용**된다. 또한 프로그래밍을 더 배우기 쉽게 하고 소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 **코드** 분석을 가능하게 하는 장점을 갖고 있다. 그러나 지나친 프로그램의 객체화 경향은 실제 세계의 모습을 그대로 반영하지 못한다는 비판을 받기도 한다.

목차 [숨기기]

- 1 역사
 - 1.1 객체 지향 언어의 시초
 - 1.2 스몰토크
 - 1.3 에이다
 - 1.4 이후에는
- 2 기본 구성 요소
- 3 특징
 - 3.1 자료 추상화
 - 3.2 상속
 - 3.3 다중 상속
 - 3.4 다형성 개념
 - 3.5 동적 바인딩
- 4 장점
- 5 객체 지향 언어

프로그래밍 패러다임

- 행위
- 에이전트 지향
- 배열 지향
- 오토마타 기반
- 병행 컴퓨팅
 - 상대주의적 프로그래밍
- 데이터 기반
- 선언형 (반의어: 명령형)
 - 함수형
 - 함수 논리형
 - 순수 함수형
 - 논리형
 - 가추법
 - Answer set
 - 병행 논리
 - 함수 논리
 - 귀납적 논리
- 제약
 - 제약 논리
 - 병행 제약 논리
- 데이터플로

2. Mark

로그인하지 않음 토론 기여 계정 만들기 로그인

읽기

편집

역사 보기

위키백과 검색



위키백과
우리 모두의 백과사전

대문

사용자 모임

요즘 화제

최근 바뀜

모든 문서 보기

임의 문서로

도움말

기부

도구

여기를 가리키는 문서

가리키는 글의 최근 바뀜

파일 올리기

특수 문서 목록

고유 링크

문서 정보

위키데이터 항목

이 문서 인용하기

다른 프로젝트

위키미디어 공용

인쇄/내보내기

책 만들기

PDF로 다운로드

인쇄용 판

다른 언어



العربية

English



Wiki Loves Monuments: 문화재 사진을 찍어 위키백과에 기여하고, 우승에도 도전해 보세요!

더 알아보기



Title1

객체 지향 프로그래밍

위키백과, 우리 모두의 백과사전.

객체 지향 프로그래밍(영어: Object-Oriented Programming, OOP)은 컴퓨터 프로그래밍의 패러다임을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 "객체"들의 모임으로 파악하고자 하는 것이다. 각각의 객체는 메시지를 주고

객체 지향 프로그래밍은 프로그램을 유연하고 변경이 용이하게 만들기 때문에 대규모 소프트웨어 개발에 많이 사용된다. 또한 프로그래밍을 더 배우기 쉽게 하고 소프트웨어 개발과 보수를 간편하게 하며, 보다 직관적인 코드 분석을 가능하게 하는 장점을 갖고 있다. 그러나 지나친 프로그램의 객체화 경향은 실제 세계의 모습을 그대로 반영하지 못한다는 비판을 받기도 한다.

목차 [숨기기]

1 역사

- 1.1 객체 지향 언어의 시초
- 1.2 스몰토크
- 1.3 에이다
- 1.4 이후에는

2 기본 구성 요소

3 특징

- 3.1 자료 추상화
- 3.2 상속
- 3.3 다중 상속
- 3.4 다형성 개념
- 3.5 동적 바인딩

4 장점

5 객체 지향 언어

Content1-1

Content1-2

Content1-3

프로그래밍 패러다임

- 행위
- 에이전트 지향
- 배열 지향
- 오토마타 기반
- 병행 컴퓨팅
 - 상대주의적 프로그래밍
- 데이터 기반
- 선언형 (반의어: 명령형)
 - 함수형
 - 함수 논리형
 - 순수 함수형
 - 논리형
 - 가추법
 - Answer set
 - 병행 논리
 - 함수 논리
 - 귀납적 논리
 - 제약
 - 제약 논리
 - 병행 제약 논리
- 데이터플로

3. Save

Document1. *

객체지향 프로그래밍

- 컴퓨터 프로그래밍 패러다임 중 하나이다.
- 여러 개의 독립 단위, 즉 ...
- 프로그램을 유연하고 변경이 용이하게 ...

How to make this work ?

- ?