

9시 시작

- 클래스 선언
- 패키지 설정

-연산자의 이해

-변수의 이해

 01_변수(Variable).pdf

1 MB

- 데이터를 변수에 저장 시킨다.
- 변수에 기록된 값(저장된 값)은 지속적으로 사용 가능하다.
- 변수는 메모리에 값을 기록한다.

- CPU = 연산만 담당
- RAM 휘발성 주 기억 장치. 변수들이 이 곳에 저장된다.
- DISC (SSD/HD) 보조 기억 장치.

-입력 버퍼 (스택 큐?)

- 비트 2진수 값 단 한 개를 저장(0 /1) 최소 저장 단위
- 비트 8개가 모여 만들어지면 하나의 바이트가 만들어진다
- 하나의 바이트가 데이터를 처리한다 (-256~256)

[변수 선언]

boolean isTure; 논리형 /참과 거짓을 true of false 를 구분한다


char ch; 문자'형' 각 글자마다 바이트를 분배한다

String str; 문자'열'

정수형
byte (1)
short (2)
int (4byte 크기)
iong (8)

실수형 (정수가 아닌 유리수 0.123~~)
float (4)
double (8?byte)

String 뺀 나머지는 기본 자료형

 자료형의 크기

Int = Integer(정수);

float의 소수점 정확도보다 double의 소수점 정확도가 두 배 이기 때문에 double이다

char == character

string == 참조형 이다.



자료형의 범위

E의 의미 == 10 ;

변수의 명령 규칙

iteral의 규칙



변수의 규칙 기억하기

안되는 것 주의하기

그냥 카멜 표기법만 쓰자 ..
그냥 카멜 표기법만 쓰자 ..
그냥 카멜 표기법만 쓰자 ..
그냥 카멜 표기법만 쓰자 ..

#절대로 변하면 안되는 수 [상수] - 항상 같은 수

형 변환

자동 형 변환

```
//자동 형 변환(auto Cast)
// - 값의 범위가 큰 자료형
// - 값의 범위가 작은 자료형의 연산시
// 작은 자료형이 큰 자료형으로 컴파일러에 의하여 자동적으로 변환하는 것.
```



강제 형 변환

- 오늘 배운 것.(0222.화)
- 이클립스 런처 이용
 - 테마 적용

- 프로젝트 폴더 생성
- 클래스 생성
- 메인 메소드 생성 `public static void main(String[] args){}`
- 연산을 원하면 쌍따옴표를 제거
- 문자열을 원하면 "쌍따옴표" 안에
- 문자열 + 숫자
- +는 이어 쓰기로 적용
- "~문자~" + (--연산--)
- 변수의 종류 학습.
- 변수의 장점들
- -가독성 증가
- -재사용성 증가
- -코드 길이의 감소
- -유지보수성의 증가
- 기본 자료형 8가지
- 논리 : boolean / 2
- 정수 : byte , short , int , long / 1 2 4 8
- 실수 : float , double / 4 8
- 문자형 : char(2byte 유니코드)
- 변수의 선언
- 변수에 대입
- 변수 초기화
- 변수에 대입되는 값을 리터럴(literal)이라 한다.
- 자료형에 따라 표기법이 있다
- long == L / float == F
- char 에는 문자열 하나가 대입 되는 게 아니라 들어간 유니코드의 자리 값이 들어간다
- 명령 문자
- 대소문자 구분
- 길이 제한 없음
- 예약어 들은 사용 금지
- 숫자로 시작은 금지
- 특수 문자 \$, _ 사용 가능하지만 사용 안 한다
- #카멜 표기법만 사용한다 생각한다 - 연결된 단어의 첫 글자만 사용한다
- 변수명은 언어를 가리지 않지만 영어를 사용한다.
- 상수 == 변하지 않는 수 == final로 선언
- 항상 변하지 않는 수
- 상수는 변수 중에 하나다
- 특정한 수에 의미를 부여하거나 PI등 변하면 안되는 수
- 형 변환
- 자동 형 변환 / 강제 형 변환
- 같은 자료형만 연산 가능
- 작은 자료형과 큰 자료형의 연산 시 큰 자료형을 기준으로 자동 변환
- 정수형 -> 실수형 순서대로 (char는 short와 같이 처리된다.)
- long 이 float으로 변환 시 값을 초과할 경우에는 오버플로우가 일어나게 됨