

과제 #01

데이터사이언스를 위한 컴퓨팅 1 (2022년도 1학기, M3239.005500)

Due: 2022년 3월 23일(수) 23시 59분

1 Compilation Process

1.1 Preprocessing

- (a) `cpp -v /dev/null`로 C preprocessor가 헤더 파일을 검색하는 위치를 알 수 있다.

```
#include <...> search starts here:
/usr/lib/gcc/x86_64-linux-gnu/7/include
/usr/local/include
/usr/lib/gcc/x86_64-linux-gnu/7/include-fixed
/usr/include/x86_64-linux-gnu
/usr/include
End of search list.
```

`/usr/include/stdio.h`, `/usr/include/math.h`에 헤더 파일이 있으며 각각 870, 1266줄이다.

- (b) `extern int scanf (const char *__restrict __format, ...) ;`
`extern int printf (const char *__restrict __format, ...);`
`extern double sqrt (double __x) __attribute__ ((__nothrow__ , __leaf__));`
- (c) 구현 없이 함수의 prototype만 들어있다. 함수의 실제 구현은 linking 때 실행파일에 포함된다.

1.2 Compilation

- (a) `gcc -c sqrt.c -o sqrt.o`

2 C++ Programming Practice

조교의 구현은 예시일 뿐이며, 다른 방법으로 구현해도 무방하다.

2.1 Concepts in OOP

2.1.1 shapes.cpp 파일의 구현

```
#include "shapes.h"
#include <stdio.h>

Shape::Shape(int x, int y) {
    x_pos = x;
    y_pos = y;
}
```

```

void Point::DrawSelf() {
    printf("Draw Point at (%d, %d)\n", x_pos, y_pos);
}

void Line::DrawSelf() {
    printf("Draw Line at (%d, %d)\n", x_pos, y_pos);
}

void Triangle::DrawSelf() {
    printf("Draw Triangle at (%d, %d)\n", x_pos, y_pos);
}

void Equilateral::DrawSelf() {
    printf("Draw Equilateral at (%d, %d)\n", x_pos, y_pos);
}

void Isosceles::DrawSelf() {
    printf("Draw Isosceles at (%d, %d)\n", x_pos, y_pos);
}

void Scalene::DrawSelf() {
    printf("Draw Scalene at (%d, %d)\n", x_pos, y_pos);
}

void Quadrilateral::DrawSelf() {
    printf("Draw Quadrilateral at (%d, %d)\n", x_pos, y_pos);
}

void Rectangle::DrawSelf() {
    printf("Draw Rectangle at (%d, %d)\n", x_pos, y_pos);
}

void Square::DrawSelf() {
    printf("Draw Sqaure at (%d, %d)\n", x_pos, y_pos);
}

```

2.1.2 shapes.h 파일의 구현

```

#pragma once

class Shape {
public:
    Shape(int x, int y);
    virtual void DrawSelf() = 0;

protected:
    int x_pos;
    int y_pos;
};

class Point : public Shape {

```

```

public:
    Point(int x, int y) : Shape(x, y) {}
    void DrawSelf();
};

class Line : public Shape {
public:
    Line(int x, int y) : Shape(x, y) {}
    void DrawSelf();
};

class Triangle : public Shape {
public:
    Triangle(int x, int y) : Shape(x, y) {}
    void DrawSelf();
};

class Equilateral : public Triangle {
public:
    Equilateral(int x, int y) : Triangle(x, y) {}
    void DrawSelf();
};

class Isosceles : public Triangle {
public:
    Isosceles(int x, int y) : Triangle(x, y) {}
    void DrawSelf();
};

class Scalene : public Triangle {
public:
    Scalene(int x, int y) : Triangle(x, y) {}
    void DrawSelf();
};

class Quadrilateral : public Shape {
public:
    Quadrilateral(int x, int y) : Shape(x, y) {}
    void DrawSelf();
};

class Rectangle : public Quadrilateral {
public:
    Rectangle(int x, int y) : Quadrilateral(x, y) {}
    void DrawSelf();
};

class Square : public Quadrilateral {
public:
    Square(int x, int y) : Quadrilateral(x, y) {}
    void DrawSelf();
};

```

};