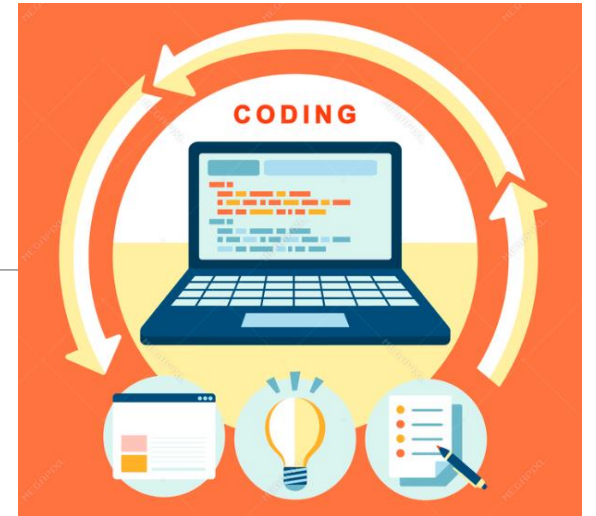


# 자료구조

실습활동



# 그래프 (Graph)

## ADT(Abstract Data Type)



# 무방향 그래프 ADT

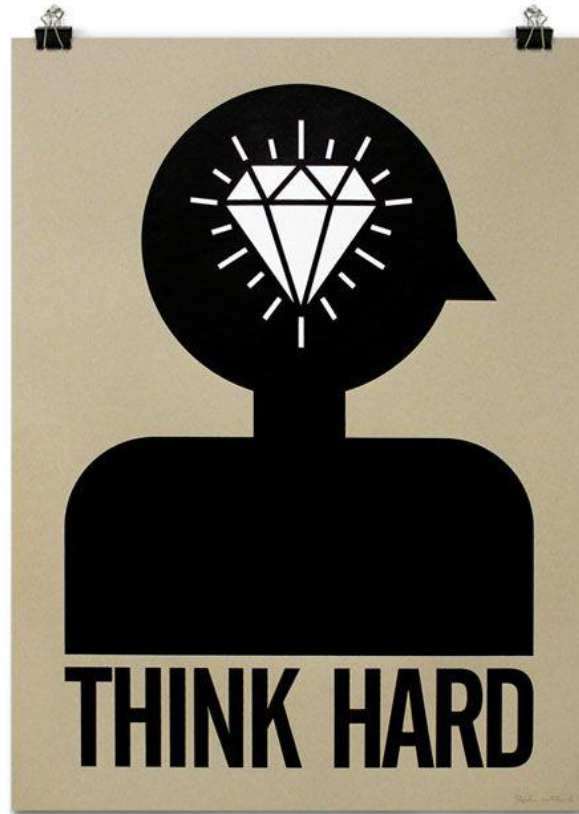
작업: ADT (구현자 관점)	명령어 (사용자 관점)	실행 결과 (자료 관점)
<code>create(my_graph)</code>	G	()
<code>insert_vertex(my_graph, vertex)</code>	+A	(A)
<code>insert_vertex(my_graph, vertex)</code>	+B	(A,B)
<code>insert_vertex(my_graph, vertex)</code>	+C	(A, B, C)
<code>insert_vertex(my_graph, vertex)</code>	+D	(A, B, C, D)
<code>add_edge(my_graph, vertex1, vertex2)</code>	E(A,B)	(A(B), B(A), C, D)

작업: ADT (구현자 관점)	명령어 (사용자 관점)	실행 결과 (자료 관점)
add_edge(my_graph, vertex1, vertex2)	E(A,C)	(A(B,C), B(A), C(A), D)
add_edge(my_graph, vertex1, vertex2)	E(A,D)	(A(B,C,D),B(A),C(A),D(A))
print(my_graph)	L	(A(B,C,D),B(A),C(A),D(A))
degree_of_vertex(my_graph, vertex)	V(A)	3
is_connected(my_graph)	C	True
is_empty(my_graph)	N	False
adjacent(my_graph, vertex)	A(A)	{B, C, D}
delete_vertex(my_graph, vertex)	-A	(B, C, D)
is_connected(my_graph)	C	False

작업: ADT (구현자 관점)	명령어 (사용자 관점)	실행 결과 (자료 관점)
※ (B, C, D)에서 (A(B,C,D),B(A),C(A),D(A)) graph를 다시 작성하는 과정을 보이시오.		
path_exist(my_graph, vertex1, vertex2)	P(B,C)	True
num_of_vertex(my_graph)	X	4
num_of_edge(my_graph)	H	3
delete_edge(my_graph,vertex1,vertex2)	D(B,C)	<b>ERROR!</b>
delete_edge(my_graph,vertex1,vertex2)	D(A,C)	(A(B,D), B(A), C, D(A))
rename_vertex(my_graph,old_v,new_v)	R(D,F)	(A(B,F), B(A), C, F(A))
clear(my_graph)	K	

# 자신만의 기능을 3개 추가해보세요!

---





# 【도전 프로그램】

- ◆ Weighted Graph (가중치 그래프)에 대한 ADT를 정리한 후 구현해보세요.

