실습 1:

-실습코드

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

-결과출력

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

-코드관련 설명

Point 클래스의 필드(변수)를 double x,y로 만듭니다. 이후 Point의 생성자를 두 개 만듭니다. 하나는 파라미터 없는 생성자이며 this래퍼런스를 이용해 x,y를 0으로 초기화 시킵니다. 이는 이후에 p1인 (0,0)이 됩니다. 하나는 double타입의 파라미터 두 개를 받는 생성자로 this래퍼런스를 이용해 x,y를 입력받은 값으로 초기화 시킵니다

public static void main(String[]args)에서 먼저 크기가 3인 배열을 만들고, 이는 Point 클래스의 객체가 되도록 합니다, 해당 배열 array[0]는 파라미터가 없는 생성자를 호출해 (0,0)을 저장합니다. 배열의 나머지 인덱스 값들은 for문과 scanner를 사용해 각각 x좌표와 y좌표를 입력받습니다.

배열의 첫번째 인덱스 array[0]를 p1, array[1]를 p2, array[2]를 p3라 할 때 서로의 모든 거리를 계산합니다. 이때 distance 메소드를 만들고 파라미터는 객체의 인자전달을 통한 배열의 특정 좌표가 됩니다. 두 좌표의 x좌표끼리 뺀 값과 y좌표끼리 뺀 값을 제곱해 더하고 루트씌우면 distance의 리턴값이 됩니다.

이러한 메소드를 사용해 p1 vs p2, p2 vs p3, p1 vs p3의 거리를 모두 구하고 출력하고, min를 사용해 이중 가장 작은 값을 shortest로 선언한뒤 출력합니다.

실습 2:

-실습 코드

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

-결과 출력

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

-코드관련 설명

Car 클래스에 필드(변수)로 carName, maxSpeed를 선언합니다. 이후 문자열 하나, double하나 이렇게 두 파라미터를 받는 생성자를 만들어서 this레퍼런스로 저장합니다.

public static void main(String[]args)에서 스캐너로 carName1, maxSpeed1을 입력받고 첫 번쨰 자동차의 객체로 저장합니다. 동일한 방법으로 carName2, maxSpeed2을 입력받고 두 번째 자동차의 객체를 저장합니다. 이후 두 객체 중 하나를 선택하기 위해 select 변수를 선언해 1또는 2를 입력받습니다. 1을 입력받으면 첫번째 자동차를, 2를 입력받으면 두번째 자동차의 속도를 바꾸는 겁니다. 이후 최대속도variation 를 입력받고, select의 입력값에 해당하는 자동차의 속도에 방금 입력받은 최대속도variation 를 더합니다. 어떻게 더하냐면,

change 메소드를 만듭니다. 파라미터는 variation이고, this레퍼런스로 저장된 maxSpeed에 variation을 더하면 됩니다.

정리하자면, select가 1일 때 car1의 change 메소드를 호출하며, 이때 variation을 파라미터로 합니다. 이후 변한 maxSpeed와 carName을 출력합니다.

Select가 2일때도 마찬가지 방법으로 하고 , 만약 select가 1또는 2가 아니라면 1과2중 고르라는 식의 문자열을 출력합니다.

이후 스캐너를 close하고, 클래스를 종료합니다.

실습 3:

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

-결과 출력

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

-코드 관련 설명

Figure 클래스 속 필드 (변수) name, radius, length, width, area를 선언합니다. 이후 생성자를 두 개 만듭니다. 하나는 문자열 name, double radius를 파라미터로 입력받고 this레퍼런스로 radius를 저장하는 생성자, 하나는 문자열 name, double length, double width를 파라미터로 입력받고 this 레퍼런스로 length와 width를 저장하는 생성자를 만듭니다.

메소드 오버로딩을 통해 getArea 메소드를 두 개 만듭니다. 하나는 radius를 파라미터로 받고 this.area에 3.14 \*radius \* radius값을 저장합니다. 이후 포매팅을 통해 해당 넓이 값을 출력합니다.

두번 째 메소드는 같은 getArea메소드 이지만 double length, double width를 파라미터로 입력받습니다. 이후 this.area에 파라미터 두 값을 곱해 저장하고, 이후 포매팅을 통해 해당 넓이 값을 출력합니다.

public static void main (String[]args)로 이제 입력을 받아봅시다. 객체 배열 a을 만든 후 크기는 4로 선언합니다. 해당 객체 배열에 스캐너를 통해 (String, double, double) 순으로 입력을 받을 겁니다. For문으로 a[i]를 저장할 것이고, 이 떄 첫 입력 문자열이 “Circle”이면, double radius를 입력으로 저장하고 해당 “Circle”, radius를 파라미터로 a[i] 객체에 생성자를 만듭니다. 만든 a[i] 에 getArea메소드를 호출해 원의넓이 값을 계산하고 출력합니다.

마찬가지 방법으로 첫 입력 문자열이 “Rectangle”이면, double length1, double width1를 입력으로 저장하고 해당“Rectangle” double length1, double width1를 a[i]객체에 생성자를 만듭니다. 만든 a[i]에 getArea메소드를 호출해 사각형의 넓이를 계산하고 출력합니다.