

## C++&Qt\_day1 과제 2 보고서

로봇 20 기 인턴

정보융합학부

2025404008

김다은

## 1. 예외 처리 알고리즘

### 1.1 입력 오류 처리 알고리즘

(숫자 입력 오류 처리 알고리즘)

1. cin 으로 데이터를 입력받는다.
  2. cin.fail()로 입력 실패 여부를 확인한다.
  3. 실패한 경우:
    - cin.clear()로 오류 플래그를 초기화한다.
    - cin.ignore(1000, 'Wn')로 입력 버퍼를 비운다.
    - 오류 메시지를 출력한다.
    - 다시 입력을 요청한다.
  4. 성공할 때까지 2-3 과정을 반복한다.
- 문자 입력 시, 최대 1000 자까지 무시하며 버퍼를 정리하고, while 루프를 통해 연속 오류 시에도 정상 입력이 들어올 때까지 반복 요청하도록 처리하였 tmqslek

### 1.2 수학적 공식을 활용한 입력 검증 알고리즘

#### 1 점 개수 확인 알고리즘

1. 사용자로부터 점의 개수를 입력받는다.
2. 입력된 값이 2 보다 작은지 확인한다.
3. 2 보다 작으면:
  - "최소 2 개의 점이 필요합니다" 메시지를 출력한다.
  - 다시 입력을 요청한다.
4. 2 이상일 때까지 반복한다.

#### 2 좌표 범위 검증 알고리즘

1. 최솟값과 최댓값을 순서대로 입력받는다.
2. 최솟값이 최댓값보다 크거나 같은지 확인한다.
3. 조건을 위반하면:
  - "최솟값은 최댓값보다 작아야 합니다" 메시지를 출력한다.
  - 최솟값과 최댓값을 다시 입력받는다.
4. 올바른 관계가 성립할 때까지 반복한다.

## 2. 무작위 점 생성 알고리즘

### 2.1 난수 생성 시스템

1. 생성자에서 `srand(time(NULL))`을 호출한다.
2. `time(NULL)`은 현재 시간을 초 단위로 반환한다.
3. 이 값을 난수 생성기의 시드로 사용한다.
4. 매 실행마다 다른 시드로 인해 다른 난수 시퀀스가 생성된다.

### 2.2 범위 내 정수 난수 생성 알고리즘

1. `rand()` 함수로 0 부터 `RAND_MAX` 까지 난수를 생성한다.
2. 생성된 난수를 원하는 범위로 변환:
  - $(\text{maxValue} - \text{minValue} + 1)$ 로 나머지 연산을 수행한다.
  - 결과에 `minValue`를 더해 최종 범위로 조정한다.
3. 공식:  $\text{minValue} + \text{rand()} \% (\text{maxValue} - \text{minValue} + 1)$

ex) `minValue=10, maxValue=50` 일 때,

- `rand() % 41` → 0~40 범위
- 최종 범위:  $10 + (0 \sim 40) \rightarrow 10 \sim 50$

### 2.3 점 배열 생성 및 출력

- 동적으로 할당된 `points` 배열을 사용한다.
- `numPoints` 만큼 반복하면서:
  - x 좌표용 난수를 지정 범위 내에서 생성
  - y 좌표용 난수를 지정 범위 내에서 생성
  - `Point(x, y)` 생성자를 통해 객체 생성
  - 배열에 저장하며 생성된 좌표를 화면에 출력

## 3. 거리 계산 알고리즘

### 3.1 유클리드 거리 공식

cpp

```
double calculateDistance(Point p1, Point p2) {  
    int dx = p1.nX - p2.nX; // x 축 거리  
    int dy = p1.nY - p2.nY; // y 축 거리
```

```

return sqrt(dx * dx + dy * dy); // 유클리드 거리
}

```

### 3.2 유클리드 거리 계산

1. 두 점  $P_1(x_1, y_1)$ ,  $P_2(x_2, y_2)$ 를 입력받는다.
2. x 축 거리를 계산한다:  $dx = x_1 - x_2$
3. y 축 거리를 계산한다:  $dy = y_1 - y_2$
4. 거리 제곱을 계산한다:  $dx^2 + dy^2$
5. 제곱근을 구하여 실제 거리를 반환한다:  $\sqrt{dx^2 + dy^2}$

- 부동소수점 계산을 위한 `sqrt()` 함수를 사용했습니다.

## 4. 최단/최장 거리 탐색 알고리즘

### 4.1 모든 점 쌍 거리 비교 알고리즘

1. 최단거리를 매우 큰 값(예: 99999.0)으로 초기화한다.
2. 최장거리를 0.0 으로 초기화한다.
3. 각 점 쌍의 인덱스를 저장할 변수들을 초기화한다.
4. 이중 반복문으로 모든 점 쌍을 생성한다:
  - 외부 루프  $i = 0$  부터  $numPoints - 1$  까지
  - 내부 루프  $j = i + 1$  부터  $numPoints - 1$  까지
5. 각 점 쌍마다:
  - 거리를 계산한다.
  - 현재 최단거리보다 작으면 최단거리와 인덱스를 업데이트한다.
  - 현재 최장거리보다 크면 최장거리와 인덱스를 업데이트한다.
6. 모든 비교가 완료되면 결과를 출력한다.

## 5. 전체 알고리즘

1. DistanceCalculator 객체를 생성한다.
  - 모든 변수를 0 이나 NULL 로 초기화하며, 난수 생성기에 현재 시간으로 시드를 설정한다.
2. 사용자로부터 필요한 정보를 입력받는다.

- 점의 개수(최소 2 개 이상)를 묻고 재검증한다.
- 좌표의 최솟값과 최댓값을 묻고 관계를 검증한다.
- 3. 입력받은 정보에 따라 무작위 점들을 생성한다.
  - 점의 개수만큼 x, y 좌표를 지정 범위 내 무작위 값으로 설정한다.
  - 생성된 점들의 좌표를 화면에 출력한다.
- 4. 모든 점 쌍 간 거리를 계산하여 최단거리와 최장거리를 탐색한다.
  - 모든 점을 한 쌍씩 비교하며 거리를 계산한다.
  - 최단거리와 최장거리를 계속 업데이트하고, 해당 점 쌍의 정보를 기억한다.
- 5. 결과를 깔끔하게 정리해서 출력한다.
  - 최단거리 및 해당 점 쌍 출력
  - 최장거리 및 해당 점 쌍 출력
- 6. 프로그램 종료 시 동적으로 할당한 메모리를 자동으로 정리한다.

#### 6. 예외 처리 결과 값

please define the number of points : abc

숫자만 입력해주세요.

please define the number of points : 1

최소 2 개의 점이 필요합니다.

please define the number of points : 4

please define minimum of coor. value : hello

숫자만 입력해주세요.

please define minimum of coor. value : 10

please define maximum of coor. value : 5

최솟값은 최댓값보다 작아야 합니다.

please define minimum of coor. value : 0

please define maximum of coor. value : 20

Generate Random points

Point 1. nX=8 , nY=15

Point 2. nX=12 , nY=3

Point 3. nX=1 , nY=17

Point 4. nX=18 , nY=9

--- Result ---

MinDist = 8.24621

Pair of Min Coor.(x,y): P1<8,15> & P2<12,3>

MaxDist = 18.02776

Pair of Max Coor.(x,y): P1<1,17> & P2<18,9>

--- Completed ---

```
dan@dan: ~/Desktop/hw2/build
please define minimum of coor. value : 0
please define maximum of coor. value : 20

Generate Random points
Point 1. nX=8 , nY=15
Point 2. nX=13 , nY=10
Point 3. nX=17 , nY=3
Point 4. nX=16 , nY=6
Point 5. nX=10 , nY=9
Point 6. nX=13 , nY=1
Point 7. nX=9 , nY=1
Point 8. nX=10 , nY=3
Point 9. nX=7 , nY=2
Point 10. nX=10 , nY=0

--- Result ---
MinDist = 1.41421
Pair of Min Coor.(x,y): P1<9,1> & P2<10,0>

MaxDist = 15.1327
Pair of Max Coor.(x,y): P1<8,15> & P2<10,0>

--- Completed ---
dan@dan:~/Desktop/hw2/build$ S
```

## 예외처리 내용

```
dan@dan: ~/Desktop/hw2/build
dan@dan:~/Desktop/hw2/build$ ./test
please define the number of points : r
숫자만 입력해주세요.
please define the number of points : -3
최소 2개의 점이 필요합니다.
please define the number of points : 2
please define minimum of coor. value : r
숫자만 입력해주세요.
please define minimum of coor. value : -2
please define maximum of coor. value : 3

Generate Random points
Point 1. nX=0 , nY=-1
Point 2. nX=0 , nY=3

--- Result ---
MinDist = 4.00000
Pair of Min Coor.(x,y): P1<0,-1> & P2<0,3>

MaxDist = 4.0000
Pair of Max Coor.(x,y): P1<0,-1> & P2<0,3>

--- Completed ---
dan@dan:~/Desktop/hw2/build$
```