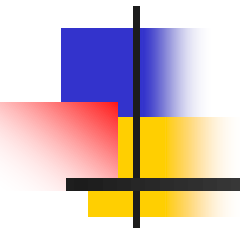


파일 처리 연습

경복대학교
소프트웨어융합과
배희호 교수

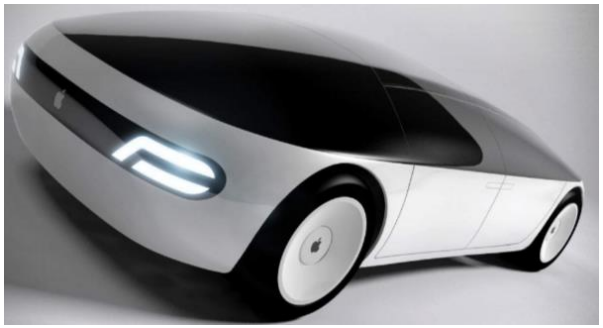
자동차 세금

- 탈것(vehicle)은 자동차(car)와 오토바이(motorcycle)로 구분하여 다음과 같이 세금(Tax)을 계산한다
  - 탈것은 금액에 따라 일정 금액의 세금을 납부함
  - 탈것은 기본 세금에 다음 조건에 따라 교육 세금(education tax)과 환경 세금(environmental tax)을 추가로 부과함
    - 연료를 가솔린(Gasoline)이나 전기(Electricity)를 사용하는 자동차는 환경 세금을 부과하지 않음
    - 연료를 디젤(Diesel)을 사용하는 자동차는 환경 세금을 추가로 부과함
    - 오토바이는 교육 세금을 부과하지 않음

자동차 세금

기본 세금
(10년 초과는 가중)

환경 세금



환경 세금
(연료: 디젤)

교육 세금

자동차 세금

- 처리 조건
  - 차종 코드

Type 1	Type 2	Type 3
승용차	승합차	오토바이

  - 연료 종료 코드

Type 1	Type 2	Type 3
Gasoline	Diesel	Electricity

  - 오토바이는 연료로 Diesel만을 사용함

자동차 세금

- 처리 조건
  - 자동차의 기본 세금은 가격에 따라 차등 부과

자동차 가격	세율
1,800,000이하	0.7%
1,800,000초과 3,600,000이하	0.8%
3,600,000초과	0.9%

  - 제조 년도가 10년이 초과한 자동차는 기본 세금의 초과한 년도의 비율로 추가 세금을 할증 부과함
    - 예) 만약 11년이 초과되었으면 기본 세금의 11%를 추가로 기본 세금에 할증함

자동차 세금

- 처리 조건
  - 오토바이의 기본 세금은 배기량에 따라 차등 부과

배기량	세율
90CC이하	0.5%
90CC초과 180CC이하	0.6%
180CC초과	0.7%

  - 제조 년도가 10년이 초과한 오토바이는 기본 세금의 초과한 년도의 비율로 추가 세금을 할증 부과함
    - 만약 11년이 초과되었으면 기본 세금의 11%를 추가로 기본 세금에 할증함

자동차 세금

- 처리 조건
  - 환경(Environment) 세금은 연료를 Diesel을 사용하는 탈 것의 기본 세금의 7%를 부과 함
  - 교육(Education) 세금은 자동차이면 기본 세금의 10%를 부과 하고, 오토바이는 기본 세금의 11%를 부과 함

자동차 세금

실행 결과

소유주	모델	제조사	년도	차종	연료	배기량	가격	세금	교육세	환경세	납부세액
정통파	SM9	삼성자동차	2021년	승용차	Electric	3,000CC	48,700,000	438,300	43,830	0	482,130
한민국	SM9	삼성자동차	2020년	승용차	Gasoline	3,000CC	46,700,000	420,300	42,030	0	462,330
자동차	SportsAge	기아자동차	1999년	승합차	Diesel	2,500CC	28,700,000	320,292	32,029	22,420	374,741
친환경	SportsAge	기아자동차	1999년	승합차	Gasoline	2,500CC	28,700,000	320,292	32,029	0	352,321
경복대	SportsAge	기아자동차	2011년	승합차	Diesel	2,500CC	28,700,000	289,296	28,929	20,250	338,475
홍길동	SONATA	현대자동차	2019년	승용차	Gasoline	1,990CC	27,500,000	247,500	24,750	0	272,250
바이든	Street10	할데이비슨	2009년	바이크	Diesel	450CC	27,800,000	221,844	0	24,402	246,246
이여주	Auto10	효성모터스	2010년	바이크	Diesel	650CC	17,800,000	140,798	0	15,487	156,285
이동국	Auto10	효성모터스	2020년	바이크	Diesel	150CC	17,800,000	106,800	0	11,748	118,548
트럼프	Street10	할데이비슨	2014년	바이크	Diesel	450CC	7,800,000	54,600	0	6,006	60,606

현재 날짜와 시간 구하기

- JAVA 8 이후
  - java.time.LocalDate
  - java.time.LocalTime
  - java.time.LocalDateTime
  - 원하는 포맷의 문자열로 변환 DateTimeFormatter 사용
- JAVA 8 이전
  - java.util.Date
  - Calendar 클래스의 getInstance() 메소드 사용
  - System 클래스의 currentTimeMillis() 메소드 사용
  - 원하는 포맷의 문자열로 변환 SimpleDateFormat 사용

현재 날짜와 시간 구하기

- LocalDate 클래스 활용 방법
  - LocalDate.now();
    - System에 default로 지정된 시간과 Time Zone존을 이용하여 현재 날짜를 가져옴
  - LocalDate.now(ZoneId.of("Europe/Paris"));
    - System 시계의 날짜를 Europe/Paris의 Time Zone을 적용하여 가져옴

```
LocalDate now = LocalDate.now();
// 연도, 월(문자열, 숫자), 일, 일(year 기준), 요일(문자열, 숫자)
int year = now.getYear();
String month = now.getMonth().toString();
int monthValue = now.getMonthValue();
int dayOfMonth = now.getDayOfMonth();
int dayOfYear = now.getDayOfYear();
String dayOfWeek = now.getDayOfWeek().toString();
int dayOfWeekValue = now.getDayOfWeek().getValue();
```

현재 날짜와 시간 구하기

- LocalTime 클래스 활용 방법
  - LocalTime.now();
    - 현재 시간을 구할 수 있음
  - DateTimeFormatter formatter = DateTimeFormatter.of Pattern("HH시 mm분 ss초");
    String formatedNow = now.format(formatter);
    - DateTimeFormatter 클래스를 이용하여 시간을 원하는 포맷의 문자열로 변환할 수 있음

현재 날짜와 시간 구하기

- LocalTime 클래스 활용 방법

```
LocalTime now = LocalTime.now();
// 현재시간 출력
System.out.println(now); // 06:25:59.985969400
// 시, 분, 초 구하기
int hour = now.getHour();
int minute = now.getMinute();
int second = now.getSecond();
// 시, 분, 초 출력
System.out.println(hour); // 6
System.out.println(minute); // 25
System.out.println(second); // 59
```

현재 날짜와 시간 구하기

- LocalDateTime 클래스 활용 방법
  - LocalDateTime.now();
    - LocalDateTime.now() 메소드를 사용하면현재 날짜와 시간을 모두 구할 수 있음
  - now.format(DateTimeFormatter.ofPattern("yyyy년 MM월 dd일 HH시 mm분 ss초"));
    - LocalDate, LocalTime 클래스의 결과를 포맷팅 했던 것과 마찬가지로DateTimeFormatter 클래스를 이용해서 포맷팅 할 수 있음

현재 날짜와 시간 구하기

- Date 클래스 활용 방법
  - 현재는 삭제 되었음
  - getYear() 메소드 사용
  - 1900년도 이후 값을 반환하므로 1900을 더해 주어야 함

```
Date time = new Date();
int year = time.getYear() + 1900;
int month = time.getMonth() + 1;
int day = time.getDate();
int hour = time.getHours();
int min = time.getMinutes();
int sec = time.getSeconds();
```

현재 날짜와 시간 구하기

- Calendar 클래스 활용 방법
  - '현재' 날짜와 시간을 구하기 위해 단 하나의 싱글톤 객체를 생성
  - getInstance() 메소드 사용
  - get() 메소드로 필요한 부분만 얻어 int 타입 변수 저장

```
Calendar cal = Calendar.getInstance();
int year = cal.get(Calendar.YEAR);
int month = cal.get(Calendar.MONTH) + 1;
int day = cal.get(Calendar.DAY_OF_MONTH);
int hour = cal.get(Calendar.HOUR_OF_DAY);
int min = cal.get(Calendar.MINUTE);
int sec = cal.get(Calendar.SECOND);
```

현재 날짜와 시간 구하기

- System.currentTimeMillis() 활용 방법
  - 1970년 1월 1일부터 경과한 시간을 long 값으로 반환
  - Milli-Second(1/1000초) 값을 반환
  - SimpleDateFormat 클래스의 fomat() 메소드 활용

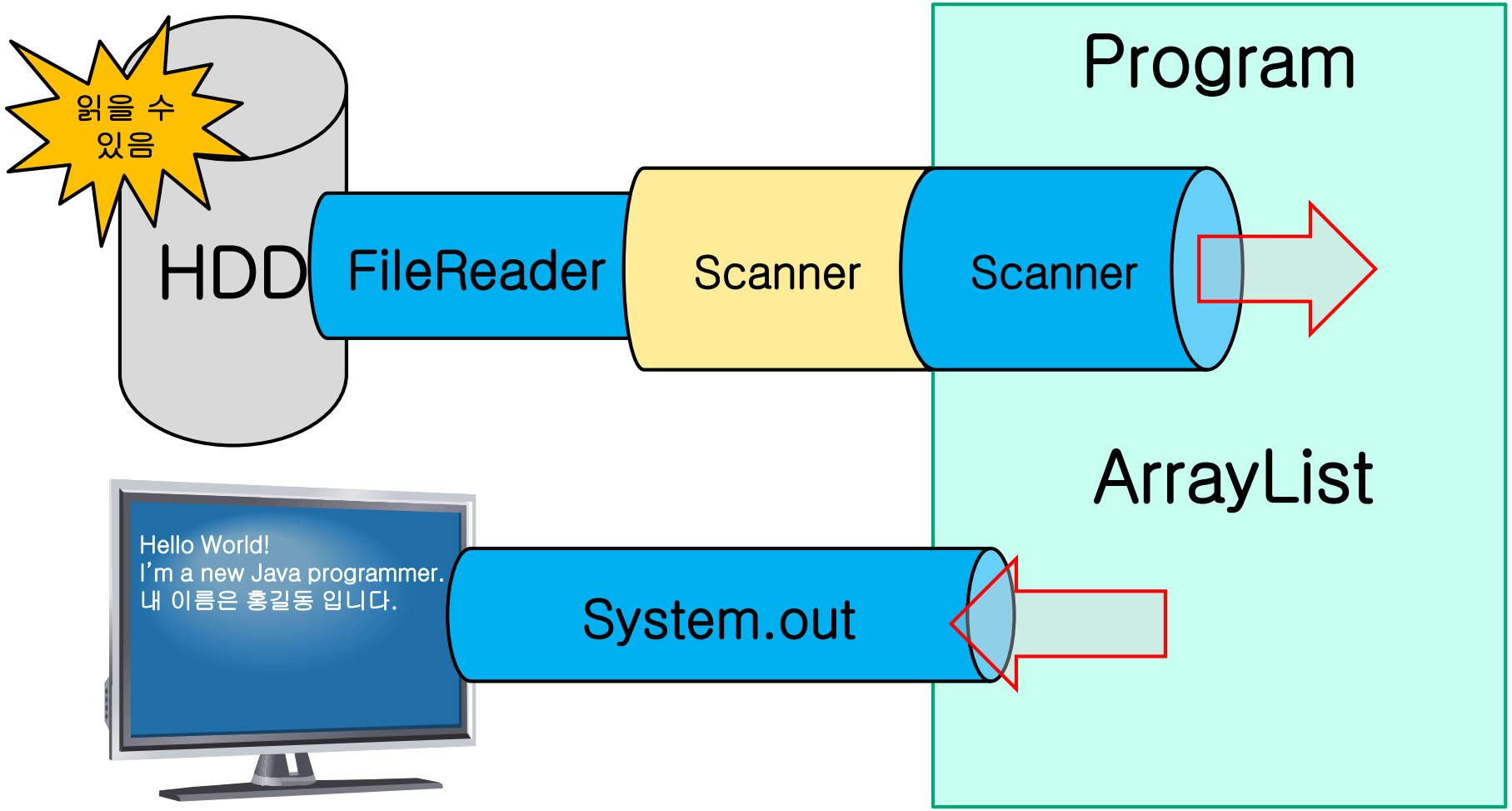
```
SimpleDateFormat format = new SimpleDateFormat ( "yyyy");
int year = Integer.parseInt(format.format(System.currentTimeMillis()));
```

현재 날짜와 시간 구하기

- SimpleDateFormat

문자	의미
y	년
M	월
d	일
D	월 구분이 없는 일(1~365)
E	요일
h	시(1~12)
H	시(0~23)
m	분
s	초
S	밀리세컨드(1/1000초)
a	오전/오후

Scanner와 Monitor 방법



자동차 세금

- Car.txt

```
1 홍길동 현대자동차 SONATA 2019 27500000 1 1 1990
1 경복대 기아자동차 SportsAge 2011 28700000 2 2 2500
1 자동차 기아자동차 SportsAge 1999 28700000 2 2 2500
1 친환경 기아자동차 SportsAge 1999 28700000 2 1 2500
2 트럼프 할데이비슨 Street10 2014 7800000 450
2 바이든 할데이비슨 Street10 2009 27800000 450
2 이동국 효성모터스 Auto10 2020 17800000 150
2 이여주 효성모터스 Auto10 2010 17800000 650
1 한민국 삼성자동차 SM9 2020 46700000 1 1 3000
1 정통파 삼성자동차 SM9 2021 48700000 1 3 3000
```

자동차 세금

- Main 클래스

```
public class Main {
    public static void main(String[] args) {
        String filename = ".//data//car.txt";
        ArrayList<Vehicle> vehicles;

        FileHandler handler = new FileHandler();
        vehicles = handler.readData(filename);

        TaxOffice tax = new TaxOffice(vehicles);
        tax.display();
    }
}
```

자동차 세금

- Vehicle 클래스

```
public abstract class Vehicle  {
    private final String owner;
    private final String manufacturer;
    private final String model;
    private final int year;

    public Vehicle(String owner, String manufacturer, String model, int year) {
        this.owner = owner;
        this.manufacturer = manufacturer;
        this.model = model;
        this.year = year;
    }

    public int getYear() {
        return year;
    }
```

자동차 세금

- Vehicle 클래스

```
    abstract public int tax();
    abstract public int environment();
    abstract public int education();

    public int total() {
       return tax() + education() + environment();
    }

    @Override
    public String toString() {
        return String.format("%3s %10.9s %6s %4d년 ",
                                                owner, model, manufacturer, year);
    }
}
```

자동차 세금

- Vehicle을 상속한 Car 클래스

```
public class Car extends Vehicle {
    private final String type;
    private final String fuelType;
    private final int engineCapacity;
    private final int price;

    public Car(String owner, String manufacturer, String model, int year,
                            int price, char type, char fuelType, int engineCapacity) {
        super(owner, manufacturer, model, year);
        this.price = price;
        if (type == '1')
            this.type = "승용차";
        else if (type == '2')
            this.type = "승합차";
        else
            this.type = "미상";
```

자동차 세금

- Vehicle을 상속한 Car 클래스

```
        if (fuelType == '1')
            this.fuelType = "Gasoline";
        else if (fuelType == '2')
            this.fuelType = "Diesel";
        else if (fuelType == '3')
            this.fuelType = "Electricity";
        else
            this.fuelType = "미상";
        this.engineCapacity = engineCapacity;
    }

    public int total() {
        return tax() + environment() + education();
    }
```

자동차 세금

- Vehicle을 상속한 Car 클래스

```
public int environment() {
    int environment = 0;
    if (fuelType.equals("Diesel"))
        environment = (int) (tax() * 7 / 100.0f);
    return environment;
}

public int education() {
    return (int) (tax() * 10 / 100.0f);
}

public int tax() {
    int tax;
    if (price <= 1800000)
        tax = (int) (price * 0.7 / 100.0f);
    else if (price <= 3600000)
        tax = (int) (price * 0.8 / 100.0f);
    else
        tax = (int) (price * 0.9 / 100.0f);
```

자동차 세금

- Vehicle을 상속한 Car 클래스

```
        Calendar cal = Calendar.getInstance();
        int temp = cal.get(Calendar.YEAR) - getYear();
        if (temp > 10) {
            tax += (int) (tax * temp / 100.0f);
        }
        return tax;
    }

    @Override
    public String toString() {
        return super.toString() +
                String.format("%5.4s %9.8s %,5dCC %,11d %,9d %,7d %,7d %,8d",
                        type, fuelType, engineCapacity, price,
                                        tax(), education(), environment(), total());
    }
}
```

자동차 세금

- Vehicle을 상속한 MotorCycle 클래스

```
public class MotorCycle extends Vehicle {
    private final int engineCapacity;
    private final int price;

    public MotorCycle(String owner, String manufacturer, String model,
                                                int year, int price, int engineCapacity) {
        super(owner, manufacturer, model, year);
        this.price = price;
        this.engineCapacity = engineCapacity;
    }

    public int total() {
        return tax() + environment();
    }

    public int environment() {
        return  (int) (tax() * 11 / 100.0f);
    }
```

자동차 세금

```
@Override
int education() {
    return 0;
}

public int tax() {
    int tax;
    if (engineCapacity <= 90)
        tax = (int)(price * 0.5f / 100.0f);
    else if (engineCapacity <= 180)
        tax = (int)(price * 0.6f / 100.0f);
    else
        tax = (int)(price * 0.7f / 100.0f);
    Calendar cal = Calendar.getInstance();
    int temp = cal.get(Calendar.YEAR) - getYear();
    if (temp > 10) {
        tax += (int)(tax * temp / 100.0f);
    }
    return tax;
}
```

자동차 세금

- Vehicle을 상속한 MotorCycle 클래스

```
    @Override
    public String toString() {
        return super.toString() +
                String.format("%5.4s %9.8s %,5dCC %,11d %,9d %,7d %,7d %,8d",
                        "바이크", "Diesel", engineCapacity, price,
                                tax(), education(), environment(), total());
    }
}
```

자동차 세금

- 세금을 징수하는 TaxOffice 클래스

```
public class TaxOffice {
    private ArrayList<Vehicle> vehicles;

    public TaxOffice(ArrayList<Vehicle> vehicles) {
        this.vehicles = vehicles;
    }

    private void sort() {
        Vehicle temp;
        for (int i = 0; i < vehicles.size() - 1; i++) {
            for (int j = i + 1; j < vehicles.size(); j++) {
                if (vehicles.get(i).total() < vehicles.get(j).total()) {
                    temp = vehicles.get(j);
                    vehicles.set(j, vehicles.get(i));
                    vehicles.set(i, temp);
                }
            }
        }
    }
```

자동차 세금

- 세금을 징수하는 TaxOffice 클래스

```
    public void display() {
        sort();
        line();
        System.out.println("소유주     모델       제조사   년도     차종     연료     배기량
                                  가격       세금    교육세   환경세   납부세액    ");
        line();
        for (int i = 0; i < vehicles.size(); i++)
            System.out.println(vehicles.get(i));
        line();
    }

    private void line() {
        System.out.println("**********************************************************
                           **********************************************************");
    }
}
```

자동차 세금

- Main 클래스

```
            new MotorCycle("이여주", "효성모터스",
                                        "Auto10", 2010, 17800000, 650),
            new Car("한민국", "삼성자동차",
                                        "SM9", 2020, 46700000, '1', '1', 3000),
            new Car("정통파", "삼성자동차",
                                      "SM9", 2021, 48700000, '1', '3', 3000)};

    TaxOffice tax = new TaxOffice(car);
    tax.display();
}
```

자동차 세금

- FileHandler 클래스

```
public class FileHandler {
    public ArrayList<Vehicle> readData(String filename) {
        ArrayList<Vehicle> vehicles = new ArrayList<>();
        File file = new File(filename);
        if (file.exists()) {
            try {
                FileReader fileReader = new FileReader(file);
                Scanner reader = new Scanner(fileReader);
                String line;
                while (reader.hasNextLine()) {
                    line = reader.nextLine();
                    Scanner lineScanner = new Scanner(line).useDelimiter(" ");
                    if (lineScanner.nextInt() == 1) {
                        String owner = lineScanner.next();
                        String manufacturer = lineScanner.next();
                        String model = lineScanner.next();
                        int year = lineScanner.nextInt();
                        int price = lineScanner.nextInt();
```

자동차 세금

- FileHandler 클래스

```
                char type = lineScanner.next().charAt(0);
                char fuelType = lineScanner.next().charAt(0);
                int engineCapacity = lineScanner.nextInt();
                vehicles.add(new Car(owner, manufacturer, model, year,
                                price, type, fuelType, engineCapacity));
            } else {
                String owner = lineScanner.next();
                String manufacturer = lineScanner.next();
                String model = lineScanner.next();
                int year = lineScanner.nextInt();
                int price = lineScanner.nextInt();
                int engineCapacity = lineScanner.nextInt();
                vehicles.add(new MotorCycle(owner, manufacturer, model,
                                                year, price, engineCapacity));
            }
            lineScanner.close();
        }
```

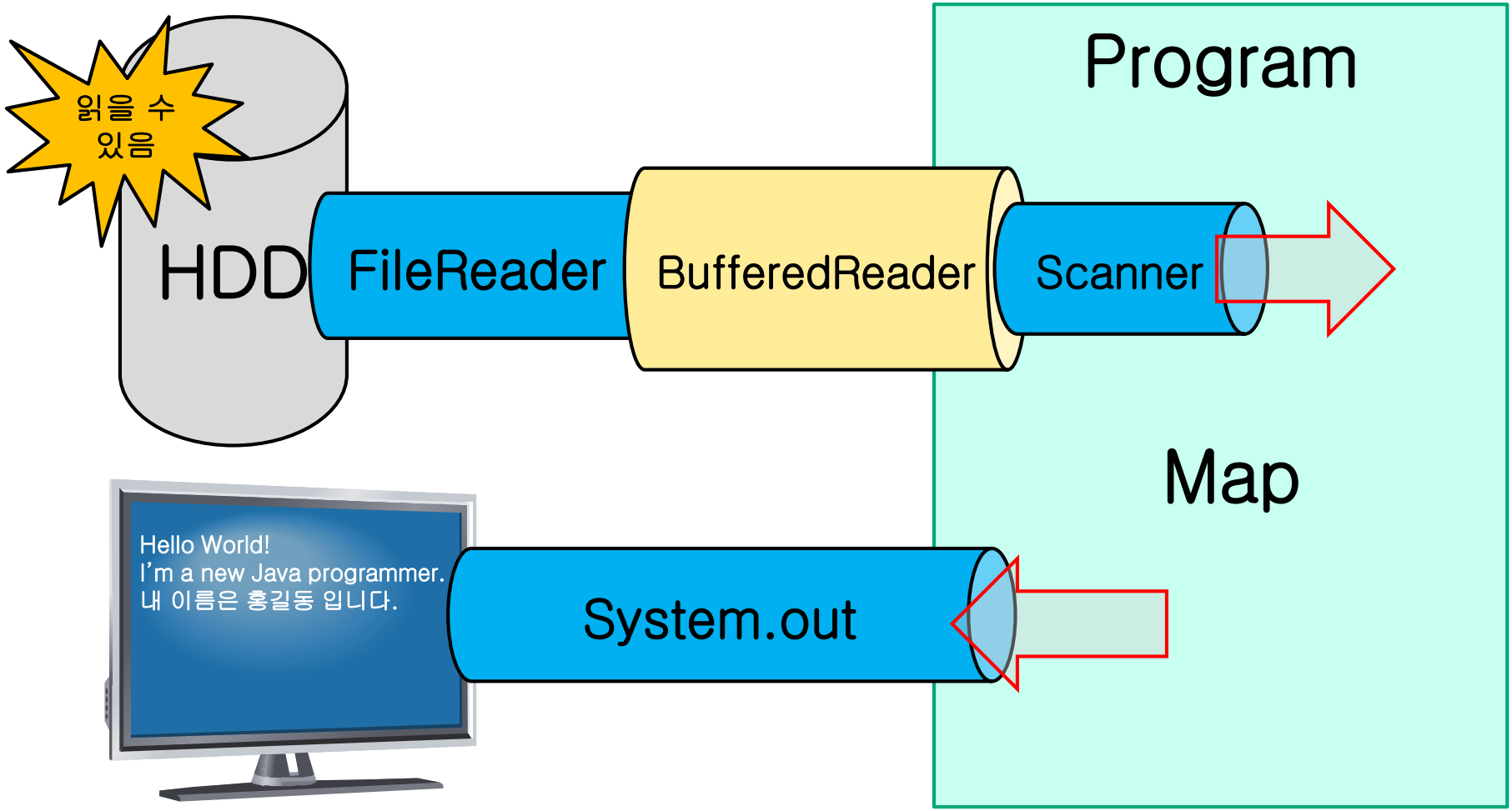
자동차 세금

- FileHandler 클래스

```
                reader.close();
                fileReader.close();
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        } else {
            System.out.println(file + "이 존재하지 않아요");
        }

        return vehicles;
    }
}
```

BufferedReader와 Monitor 방법



자동차 세금(Map)

- Main.JAVA

```
public class Main {
    public static void main(String[] args) {
        String filename = ".//data//car.txt";
        Map<Integer, Vehicle> vehicles;
        FileHandler handler = new FileHandler();
        vehicles = handler.dataRead(filename);

        TaxOffice tax = new TaxOffice(new ArrayList<>(vehicles.entrySet()));
        tax.display();
    }
}
```

자동차 세금(Map)

- TaxOffice.JAVA

```
public class TaxOffice {
    private List<Map.Entry<Integer, Vehicle>> vehicles;
    public TaxOffice(List<Map.Entry<Integer, Vehicle>> vehicles) {
        this.vehicles = vehicles;
    }
    public void sort() {
        Collections.sort(vehicles, new Comparator<Map.Entry<Integer, Vehicle>>() {
            @Override
            public int compare(Map.Entry<Integer, Vehicle> o1,
                               Map.Entry<Integer, Vehicle> o2) {
                if (o1.getValue().total() > o2.getValue().total()) {
                    return -1;
                } else if (o1.getValue().total() < o2.getValue().total()) {
                    return 1;
                }
                return 0;
            }
        });
    }
```

자동차 세금(Map)

- TaxOffice.JAVA

```
    public void display() {
        sort();
        line();
        System.out.println("소유주     모델       제조사  년도     차종     연료
                    배기량      가격       세금    교육세  환경세  납부세액    ");
        line();
        for (int i = 0; i < vehicles.size(); i++)
            System.out.println(vehicles.get(i).getValue().toString());
        line();
    }

    private void line() {
        System.out.println("**********************************************
                **********************************************");
    }
}
```

자동차 세금(Map)

- Filehandler.JAVA

```
public class FileHandler {

    public Map<Integer, Vehicle> dataRead(String filename){
        Map<Integer, Vehicle> vehicles = new HashMap<>();
        int count = 0;
        try {
            FileReader fileReader = new FileReader(filename);
            BufferedReader reader = new BufferedReader(fileReader);
            String temp;
            while ((temp = reader.readLine()) != null) {
                Scanner line = new Scanner(temp).useDelimiter(" ");
                if (line.nextInt() == 1) {
                    String owner = line.next();
                    String manufacturer = line.next();
                    String model = line.next();
                    int year = line.nextInt();
                    int price = line.nextInt();
```

자동차 세금(Map)

- Filehandler.JAVA

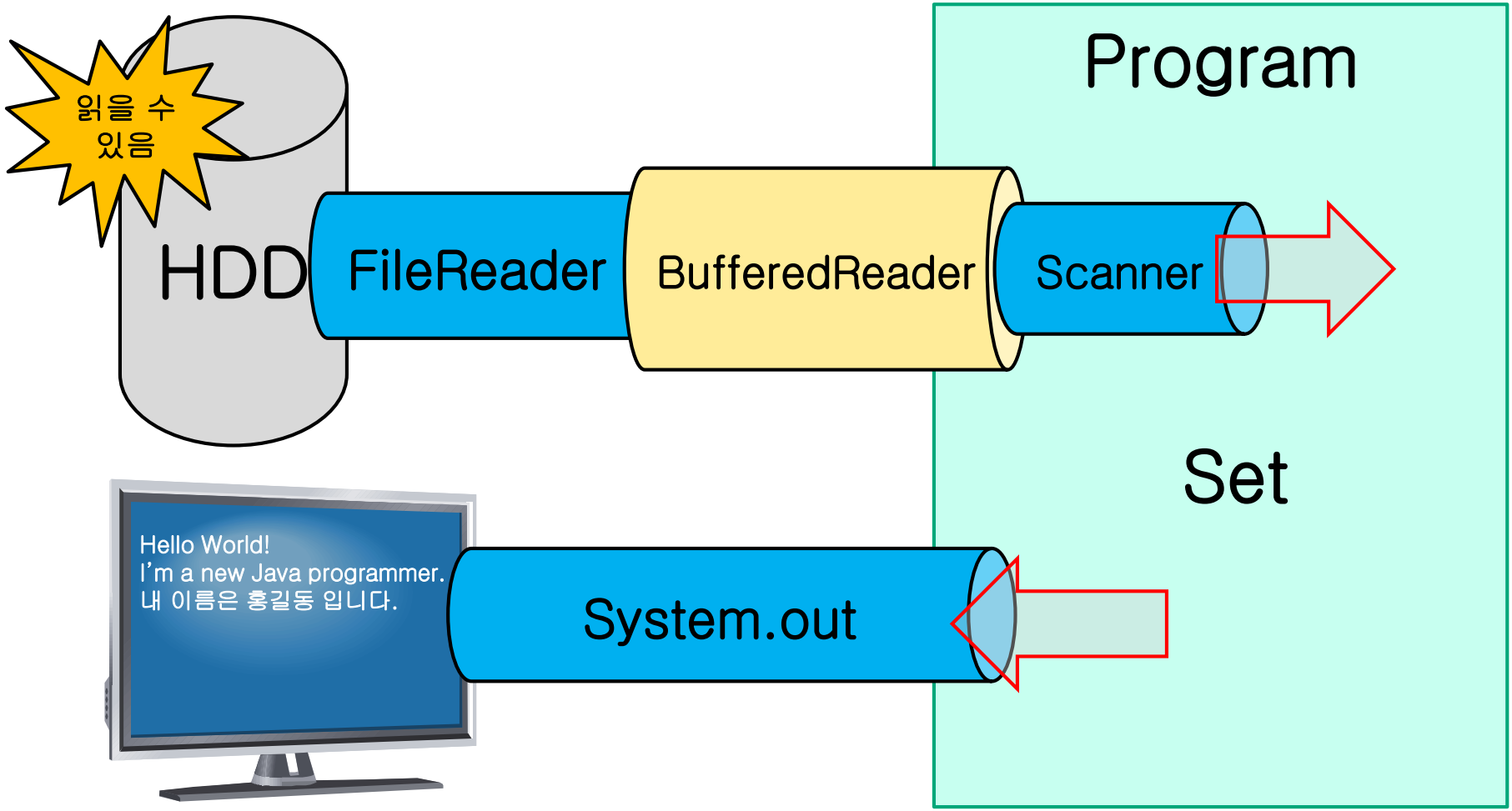
```
            char type = line.next().charAt(0);
            char fuelType = line.next().charAt(0);
            int engineCapacity = line.nextInt();
            vehicles.put(count, new Car(owner, manufacturer, model, year,
                                price, type, fuelType, engineCapacity));
        } else {
            String owner = line.next();
            String manufacturer = line.next();
            String model = line.next();
            int year = line.nextInt();
            int price = line.nextInt();
            int engineCapacity = line.nextInt();
            vehicles.put(count, new MotorCycle(owner, manufacturer, model,
                        year, price, engineCapacity));
        }
        count++;
        line.close();
    }
```

자동차 세금(Map)

- Filehandler.JAVA

```
            if (count == 0) {
                System.out.println("데이터가 없습니다");
                System.exit(-1);
            } else
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n", count);
            reader.close();
            fileReader.close();
        } catch (NullPointerException | IOException e) {
            System.out.println("오류 입니다");
            System.exit(-1);
        }
        return vehicles;
    }
}
```

BufferedReader와 Monitor 방법



자동차 세금(Set)

- Main.JAVA

```
public class Main {
    public static void main(String[] args) {
        String filename = ".//data//car.txt";
        Set<Vehicle> vehicles;
        FileHandler handler = new FileHandler();
        vehicles = handler.dataRead(filename);

        TaxOffice tax = new TaxOffice(vehicles);
        tax.display();
    }
}
```

자동차 세금(Set)

- TaxOffice.JAVA

```
public class TaxOffice {
    private final List<Vehicle> vehicles;

    public TaxOffice(Set<Vehicle> vehicles) {
        this.vehicles = new ArrayList<>(vehicles);
    }

    public void sort() {
        Descending descending = new Descending();
        vehicles.sort(descending);
    }

    private static class Descending implements Comparator<Vehicle> {
        @Override
        public int compare(Vehicle o1, Vehicle o2) {
            return Integer.compare(o2.total(), o1.total());
        }
    }
```

자동차 세금(Set)

- TaxOffice.JAVA

```
    public void display() {
        sort();
        line();
        System.out.println("소유주     모델       제조사   년도     차종     연료
                  배기량      가격       세금    교육세   환경세   납부세액    ");
        line();
        for (int i = 0; i < vehicles.size(); i++)
            System.out.println(vehicles.get(i));
        line();
    }

    private void line() {
        System.out.println("**********************************************************
                ***********************************************************");
    }
}
```

자동차 세금(Set)

Filehandler.JAVA

```
public class FileHandler {

    public Set<Vehicle> dataRead(String filename){
        Set<Vehicle> vehicles = new HashSet<>();
        try {
            FileReader fileReader = new FileReader(filename);
            BufferedReader reader = new BufferedReader(fileReader);
            String temp;
            while ((temp = reader.readLine()) != null) {
                Scanner line = new Scanner(temp).useDelimiter(" ");
                if (line.nextInt() == 1) {
                    String owner = line.next();
                    String manufacturer = line.next();
                    String model = line.next();
                    int year = line.nextInt();
                    int price = line.nextInt();
```

자동차 세금(Set)

- Filehandler.JAVA

```
            char type = line.next().charAt(0);
            char fuelType = line.next().charAt(0);
            int engineCapacity = line.nextInt();
            vehicles.add(new Car(owner, manufacturer, model, year, price,
                                 type, fuelType, engineCapacity));
        } else {
            String owner = line.next();
            String manufacturer = line.next();
            String model = line.next();
            int year = line.nextInt();
            int price = line.nextInt();
            int engineCapacity = line.nextInt();
            vehicles.add(new MotorCycle(owner, manufacturer, model, year,
                                        price, engineCapacity));
        }
        line.close();
    }
```

자동차 세금(Set)

- Filehandler.JAVA

```
            if (vehicles.size() == 0) {
                System.out.println("데이터가 없습니다");
                System.exit(-1);
            } else
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n",
                                                        vehicles.size());
            reader.close();
            fileReader.close();
        } catch (NullPointerException | IOException e) {
            System.out.println("오류 입니다");
            System.exit(-1);
        }
        return vehicles;
    }
}
```

자동차 세금(Queue)

- Main.JAVA

```
public class Main {
    public static void main(String[] args) {
        String filename = ".//data//car.txt";
        Queue<Vehicle> vehicles;
        FileHandler handler = new FileHandler();
        vehicles = handler.dataRead(filename);

        TaxOffice tax = new TaxOffice(vehicles);
        tax.display();
    }
}
```

자동차 세금(Queue)

- FileHandler.JAVA

```
public class FileHandler {

    public Queue<Vehicle> dataRead(String filename) {
        Queue<Vehicle> vehicles = new LinkedList<>();

        try {
            FileReader fileReader = new FileReader(filename);
            BufferedReader reader = new BufferedReader(fileReader);
            String temp;
            while ((temp = reader.readLine()) != null) {
                Scanner line = new Scanner(temp).useDelimiter(" ");
                if (line.nextInt() == 1) {
                    String owner = line.next();
                    String manufacturer = line.next();
                    String model = line.next();
                    int year = line.nextInt();
                    int price = line.nextInt();
```

자동차 세금(Queue)

- FileHandler.JAVA

```
                char type = line.next().charAt(0);
                char fuelType = line.next().charAt(0);
                int engineCapacity = line.nextInt();
                vehicles.add(new Car(owner, manufacturer, model, year, price,
                                type, fuelType, engineCapacity));
            } else {
                String owner = line.next();
                String manufacturer = line.next();
                String model = line.next();
                int year = line.nextInt();
                int price = line.nextInt();
                int engineCapacity = line.nextInt();
                vehicles.add(new MotorCycle(owner, manufacturer, model, year,
                                price, engineCapacity));
            }
            line.close();
        }
```

자동차 세금(Queue)

- FileHandler.JAVA

```
            if (vehicles.size() == 0) {
                System.out.println("데이터가 없습니다");
                System.exit(-1);
            } else {
                System.out.printf("데이터를 성공적으로 %d개 읽었습니다₩n",
                                                        vehicles.size());
            }
            reader.close();
            fileReader.close();
        } catch (NullPointerException | IOException e) {
            System.out.println("오류 입니다");
            System.exit(-1);
        }
        return vehicles;
    }
}
```

자동차 세금(Queue)

- TaxOffice.JAVA

```
public class TaxOffice {
    private final Queue<Vehicle> vehicles;

    public TaxOffice(Queue<Vehicle> vehicles) {
        this.vehicles = vehicles;
    }

    public void sort() {
        List<Vehicle> vehicleList = new ArrayList<>(vehicles);
        vehicleList.sort(new Descending());
        vehicles.clear();
        vehicles.addAll(vehicleList);
    }
    private static class Descending implements Comparator<Vehicle> {
        @Override
        public int compare(Vehicle o1, Vehicle o2) {
            return Integer.compare(o2.total(), o1.total());
        }
    }
```

자동차 세금(Queue)

- TaxOffice.JAVA

```
public void display() {
    sort();
    line();
    System.out.println("소유주    모델      제조사  년도    차종    연료
                배기량     가격      세금   교육세  환경세  납부세액   ");
    line();
    Iterator<Vehicle> iterator = vehicles.iterator();
    while (iterator.hasNext()) {
        Vehicle vehicle = iterator.next();
        System.out.println(vehicle);
    }
    line();
}
private void line() {
    System.out.println("**********************************************
            **********************************************");
}
}
```