SAT: Conventions

Improve the code quality of the following programs:

1.

```
class Rover {
    static final double WalkingSpeed = 3;

    final String SerialNumber;
    double MilesPerHour;

    Rover(String NewSerialNumber) {
        SerialNumber = NewSerialNumber;
    }

    void Drive() {
        MilesPerHour = WalkingSpeed;
    }
    void Stop() {
        MilesPerHour = 0;
    }
}
```

```
class Rover {
    static final double WALKING_SPEED = 3;

    final String serialNumber;
    double milesPerHour;

    Rover(String serialNumber) {
        this.serialNumber = serialNumber;
    }

    void drive() {
        milesPerHour = WALKING_SPEED;
    }

    void stop() {
        milesPerHour = 0;
    }
}
```

2.

```
class Astronaut {

    String name;
    boolean retired;

    Astronaut(String name) {
        this.name = name;
    }

    String getFullName() {
        return name;
    }

    void setFullName(String name) {
        this.name = name;
    }

    boolean getRetired() {
        return retired;
    }

    void setRetiredState(boolean retired) {
        this.retired = retired;
    }
}
```

```
class Astronaut {
    private String name;
    private boolean retired;

    public Astronaut() {
    }

    public Astronaut(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public boolean isRetired() {
        return retired;
    }

    public void setRetired(boolean retired) {
        this.retired = retired;
    }

}
```

3.

```
class Inventory {
    List<Supply> sl = new ArrayList<>();

    boolean isInStock(String n) {
        Supply s = new Supply(n);
        int l = 0;
        int h = sl.size() - 1;

        while (l <= h) {
            int m = l + (h - l) / 2;
            int c = sl.get(m).compareTo(s);

            if (c < 0) {
                l = m + 1;
            } else if (c > 0) {
                h = m - 1;
            } else {
                return true;
            }
        }

        return false;
    }
}
```

```
class Inventory {
    List<Supply> sortedList = new ArrayList<>();

    boolean isInStock(String name) {
        Supply supply = new Supply(name);
        int low = 0;
        int high = sortedList.size() - 1;

        while (low <= high) {
            int middle = low + (high - low) / 2;
            int comparison = sortedList.get(middle).compareTo(supply);

            if (comparison < 0) {
                low = middle + 1;
            } else if (comparison > 0) {
                high = middle - 1;
            } else {
                return true;
            }
        }

        return false;
    }
}
```

4.

```
class Logbook {
    static final Path DIR = Paths.get("/var/log");
    static final Path CSV = DIR.resolve("stats.csv");
    static final String GLOB = "*.log";

    void createStats() throws IOException {
        try (DirectoryStream<Path> dirStr =
                     Files.newDirectoryStream(DIR, GLOB);
             BufferedWriter bufW = Files.newBufferedWriter(CSV)) {
            for (Path lFile : dirStr) {
                String csvLn = String.format("%s,%d,%s",
                        lFile,
                        Files.size(lFile),
                        Files.getLastModifiedTime(lFile));
                bufW.write(csvLn);
                bufW.newLine();
            }
        }
    }
}
```

```
class Logbook {
    static final Path LOG_FOLDER = Paths.get("/var/log");
    static final Path STATISTICS_CSV = LOG_FOLDER.resolve("stats.csv");
    static final String FILE_FILTER = "*.log";

    void createStatistics() throws IOException {
        try (DirectoryStream<Path> logs =
                     Files.newDirectoryStream(LOG_FOLDER, FILE_FILTER);
             BufferedWriter writer =
                     Files.newBufferedWriter(STATISTICS_CSV)) {
            for (Path log : logs) {
                String csvLine = String.format("%s,%d,%s",
                        log,
                        Files.size(log),
                        Files.getLastModifiedTime(log));
                writer.write(csvLine);
                writer.newLine();
            }
        }
    }
}
```