Плюсы:

- Наличие организации проекта в виде классов
- Не используется множественное наследование
- В основном в коде содержатся «говорящие» названия классов, методов
- Именование констант, методов, классов выдержано в едином стиле
- Реализация методов эффективная, используются стандартные структуры данных
- Значимые комментарии, возможно, есть, но их невозможно прочитать
- Отсутствуют закомментированные участки кода

Минусы:

Комментарии. Без комментариев.

• Встречаются длинные конструкции, что снижает читабельность кода

```
out.close();
} catch (IOException e) {

JOptionPane.showMessageDialog(null, "Îøèáêà çàïèñè â ôàéë " + SAVE_FILE, "Îøèáêà", JOptionP.

}

break;

case KeyEvent.VK_L:

if (ctrlHold) {

FileInputStream fileIn = null;

try {

fileIn = new FileInputStream(SAVE_FILE);

} catch (FileNotFoundException el) {

JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêàà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêàà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà " + SAVE_FILE, "Îøèáêà", JOptionPane.showMessageDialog(null, "Îøèáêà çàãóóçèè èç ôàéëà")
```

 Местами используются краткие неинформативные имена переменных, также изредка используются необъявленные «магические» константы

```
if ( !Panel.getCurrentSimScreen()._waypoints.isEmpty() && Panel.getCurrentSimScreen()._persons.size() < peopleLimit
    // Iôiôiā Iî âñå òî÷êàì ôåñIàóià
    for (WayPoint point : Panel.getCurrentSimScreen()._waypoints) {
        Random r2 = new Random();
        double rd = r2.nextDouble();
        if (point.isSpawn && rd < 0.01) {
            Person p = new Person(new Point2D.Double(point.getP().x, point.getP().y));
        }
}</pre>
```

Использование глобальных переменных класса

```
// TODO: çàïðåòèòü ðåäàêòèðîâàíèå âî âðåìÿ èìèòàöèè
public final static int winModeWidth = 1150, winModeHeight = 600;  // øèðèíà, âûñîòà î
public static boolean altHold, shiftHold, m1Hold, ctrlHold;
public static boolean drawAI;
public static boolean paused;
public static int objectAdding = 0;  // êàêîé îpúåêò äîáàâëÿåòñÿ
public static int peopleLimit = 0;
```

• Используются методы большого размера

```
public void actionPerformed(ActionEvent evt) {
        // TODO: ãåíåðèðîâàòü ïóòü, åñëè íåò íàçíà÷åíèÿ è ïóòè
       // áðàòü ñëåäóþùóþ òî÷êó èç ïóòè, åñëè ñåé÷àñ â òî÷êå íàçí.
       // äëÿ îáõîäà ñîçäàòü boolean[] ðàçìåðà _waypoints.size()
       // Collections.shuffle(Panel.getCurrentSimScreen()._waypoints);
       if ( !Panel.getCurrentSimScreen()._waypoints.isEmpty() && Panel.getCurrentSimScreen()._persons.size() < peopleLimit
               // ïðîõîä ïî âñå òî÷êàì ðåñïàóíà
               for (WavPoint point : Panel.getCurrentSimScreen(), wavpoints) {
                       Random r2 = new Random();
                       double rd = r2.nextDouble();
                       if (point.isSpawn && rd < 0.01) {
                               Person p = new Person(new Point2D.Double(point.getP().x, point.getP().y));
                               // åñëè íàéäåí íîâûé ïóòü, äîáàâèòü \divåëîâåêà
                               if (Logic.FindRandomPathToAnywhere(p, point, true))
                                   Panel.getCurrentSimScreen()._persons.add(p);
                       }
        if (!Panel.getCurrentSimScreen()._persons.isEmpty())
                for (int i = 0; i < Panel.getCurrentSimScreen()._persons.size(); i++) {</pre>
                       Person person = Panel.getCurrentSimScreen()._persons.get(i);
                       Point2D.Double curPos = person.getPosition();
                       double dx = person.goingTo.getP().x - curPos.x;
                                                                        И далее
                       double dy = person.goingTo.getP().y - curPos.y;
```

• Используются сгенерированные методы и названия объектов

Оценка: 6/10