Ревью на PopovaKharakhordinPI42 от ZayvyPolomoshnovSergeevPI42

Плюсы:

+1 Хорошая комментированность кода

+2 Информативное название полей (в большинстве случаев)

```
    private Point2D.Double position;    // позиция
    private int size;           // размер (кружка?)
    private int speed;          // скорость перемещения

    boolean hasCloth;           // есть куртка с собой
    boolean coatRoomCheck;      // сдавал куртку в гардероб
    boolean wantsToEat;         // хочет ли есть
    boolean wantsToDrink;       // хочет ли пить
    int toPrint;                // сколько листов нужно распечатать

    double wannaEatProb;        // вероятность, что захочет есть
    double wannaDrinkProb;      // вероятность, что захочет пить
    double wannaLeave;          // вероятность, что пойдет на выход

    public int cash;            // сколько имеет денег наличкой
    public int credit;          // сколько имеет денег на карте
    public WayPoint goingTo;    // к какой точке сейчас идет
    public ArrayList<WayPoint> path;    // путь, по которому идет
    public ArrayList<String> log;       // события, произошедшие с че

    public WayPoint backToStore;        // назад от банкомата

    public boolean isSelected;          // выделен ли
    public boolean isWorker;            // работник ли
    public UsableMapObject object;      // объект работника
```

+3 Отделение логики работы приложения от его графической части

+4 Наличие TODO-комментариев

Минусы:

-1 Длинные последовательности символов (строки) Юникода без обозначения их содержания на русском (строки представляют русские слова).

```
wantsToEat = new JCheckBox("\u0445\u043E\u0447\u0435\u0442 \u0435\u0441\u0442\u044C");
wantsToEat.setEnabled(false);
wantsToEat.setBounds(368, 11, 200, 23);
frame.getContentPane().add(wantsToEat);

wantsToDrink = new JCheckBox("\u0445\u043E\u0447\u0435\u0442 \u043F\u0438\u0442\u044C");
wantsToDrink.setEnabled(false);
wantsToDrink.setBounds(368, 37, 200, 23);
frame.getContentPane().add(wantsToDrink);

hasCloth = new JCheckBox("\u0435\u0441\u0442\u044C \u043A\u0443\u0440\u0442\u043A\u0430 \u0441 \u0441\u043E\u0431
    \u043E\u0439");
hasCloth.setEnabled(false);
hasCloth.setBounds(368, 63, 200, 23);
frame.getContentPane().add(hasCloth);

checkCoatRoom = new JCheckBox("\u0441\u0434\u0430\u0432\u0430\u043B \u043A\u0443\u0440\u0442\u043A\u0443 \u0432
    \u0433\u0430\u0440\u0434\u0435\u0440\u043E\u0431");
```

-2 Местами ширина кода достигает 160 и более символов (отсутствует перенос строк), что усложняет чтение и восприятие.

```
// отправить работников, если мало товара
if (!Panel.getCurrentSimScreen()._objects.isEmpty() && Panel.getCurrentSimScreen()._persons.size() < peopleLimit)
    for (MapObject go : Panel.getCurrentSimScreen()._objects)
        if (go != null && go.getClass().getSimpleName().equals("UsableMapObject")) {
            UsableMapObject ugo = (UsableMapObject) go;
            Random r2 = new Random();
            // пополнение автомата и магазина
            if ( ((ugo.getHas() < ugo.getCapacity()*0.2 && (ugo.getType() == 0 || ugo.getType() == 4) && ugo.admin == null) ||
                    (ugo.getHas() < Logic.minPaperLimit && ugo.getType() == 1 && ugo.admin == null)) && r2.nextDouble() < Logic.goingToTheWork ) {
```

-3 Использование оператора if, чтобы вернуть значение логического выражения (return Выражение)

```
public static boolean checkRectToPointCollision(Rectangle rect, Point point) {
    if (rect.contains(point))
        return true;
    else
        return false;
}

public static boolean checkCircleToPointCollision(Point circle, int r, Point point) {
    int distance = (int) Math.sqrt(Math.pow((point.getX() - circle.getX()), 2) + Math.pow((point.getY() - circle.getY()), 2));
    if (distance < r)
        return true;
    else
        return false;
}
```

-4 Блоки case большого размера

```
case KeyEvent.VK_DELETE:
    if (!Panel.getCurrentSimScreen()._objects.isEmpty() && shiftHold) {
        ListIterator<MapObject> it = Panel.getCurrentSimScreen()._objects.listIterator();
        while (it.hasNext()) {
            Point cursor = MouseInfo.getPointerInfo().getLocation();
            SwingUtilities.convertPointFromScreen(cursor, frame);
            MapObject mo = it.next();
            if (mo == null) continue;
            if ( mo.getClass().getSimpleName().equals("Wall")) {
                Wall w = (Wall) mo;
                // проверяем, пересекается ли
                Rectangle rect= new Rectangle(w.getP());
                rect.add(w.getWallEnd());

                if (Panel.checkRectToPointCollision(
                        rect,
                        cursor))
                    it.remove();
            }
            if ( mo.getClass().getSimpleName().equals("UsableMapObject")) {
                UsableMapObject vm = (UsableMapObject) mo;
                // проверяем, пересекается ли
                Rectangle rect = new Rectangle(new Point(vm.getP().x - Panel.UsableObjectWidth/2, vm.getP().y - Panel.
                    UsableObjectHeight/2));
                rect.add(new Point(vm.getP().x + Panel.UsableObjectWidth/2, vm.getP().y + Panel.UsableObjectHeight/2));

                if (Panel.checkRectToPointCollision(
                        rect,
                        cursor)) {
                    if (!Panel.getCurrentSimScreen()._waypoints.isEmpty())
                        for (WayPoint wp : Panel.getCurrentSimScreen()._waypoints)
                            if (wp.getClass().getSimpleName().equals("UsableWayPoint")) {
                                UsableWayPoint vwp = (UsableWayPoint) wp;
                                if (vwp.object == vm) {
                                    Panel.getCurrentSimScreen()._waypoints.remove(vwp);
                                    break;
                                }
                            }

                    it.remove();

                }
            }
        }
    }
```

Итоговая оценка: 4- 4+ -> 5/10