

Анализ исходного кода программы **CourseWork**

В ходе анализа исходного кода данной приложения были выделены следующие достоинства и недостатки.

Достоинства:

1. Организация проекта в виде классов.
2. Графический интерфейс должен быть отделен от методов, реализующих логику системы.
3. Код выдержан в едином стиле написания.
4. Используются стандартные структуры данных.
5. Используются разработанные и принятые сообществом сторонние компоненты.
6. Отсутствие закомментированных участков кода.

Недостатки:

1. Полностью отсутствуют комментарии к методам и интерфейсам.
2. Использование неговорящих названий имен переменных.

```
Patient pa = list.getElementAt(i);
```

```
Disease d = new FormBuilder<Disease>(new Disease(), FormType.NEW).show();
```

```
Patient a = new FormBuilder<Patient>(frame.getPatients()).getElementAt(f
```

```
AbstractMeans m = null;
```

```
Object o = new FormBuilder<AbstractMe
```

3. Нечитабельное форматирование исходного кода

```
    if(f.getName().endsWith(".dat")) return f; else return new File(f.getAbsolutePath() + ".dat");  
    }  
    return null;
```

```
        if(means[i][j] != null && c.isInstance(means[i][j])) list.add((T)means[i][j]);  
    return list;
```

```
for(AbstractMeans m : a) means[m.getX()][m.getY()] = m;
```

```
id = in.readInt(); if(id > maxId) maxId = id + 1;
```

4. Магические константы

```
private static AbstractMeans[][] means = new AbstractMeans[2][18];
```

```
means = new AbstractMeans[2][18];
```

```
Main.getEventListener().onCanvasClick(e.getY() / 35, e.getX() / 35);
```

```
g.drawLine(0, 35, width - 1, 35);
```

```
for(int i = 0; i < 17; i++) g.drawLine((i + 1) * 35, 0, (i + 1) * 35, height - 1);
```

```
g.setColor(new Color(m.isSelected() ? 0xb3eca4 : 0xf0f6fd));
g.fillRect(y * 35 + 1, x * 35 + 1, 34, 34);

Image img = Utils.readImage(m.getIconLocation());
if(img != null) g.drawImage(img, y * 35 + 2, x * 35 + 2, null);
```

```
g.setColor(Color.WHITE);
g.fillRect(y * 35 + 1, x * 35 + 1, 34, 34);
```

```
public int getResult() {
    if(result == 2) return -1;
    return result;
}
```

5. Огромный блок оператора switch (около 100 строк) для обработки событий формы. Рациональней было бы использование eventListener-интерфейса.

```
switch(e.getActionCommand()) {
    case "menu-exit":
        System.exit(0);
        break;
    case "disease-selection":
    case "patient-selection":
        onListSelectionChanged((JList<?>)e.getSource());
        break;
    case "disease-add":
        Disease d = new FormBuilder<Disease>(new Disease(), FormType.NEW).show();
        if(d != null) Main.getMainFrame().getDiseases().addElement(d);
        break;
    case "disease-edit":
        Disease o = new FormBuilder<Disease>(frame.getDiseases().getElementAt(frame.getDiseasesList().getSelectedIndex()), FormType.EDIT).show();
        if(o != null) frame.getDiseasesList().repaint();
        break;
    case "disease-remove":
        if(confirmDelete()) frame.getDiseases().remove(frame.getDiseasesList().getSelectedIndex());
        break;
    case "patient-add":
        Patient p = new FormBuilder<Patient>(new Patient(), FormType.NEW).show();
        if(p != null) Main.getMainFrame().getPatients().addElement(p);
        break;
    case "patient-edit":
        Patient a = new FormBuilder<Patient>(frame.getPatients().getElementAt(frame.getPatientsList().getSelectedIndex()), FormType.EDIT).show();
        if(a != null) frame.getPatientsList().repaint();
        break;
    case "patient-remove":
        if(confirmDelete()) frame.getPatients().remove(frame.getPatientsList().getSelectedIndex());
        break;
}
```

Подсчитаем оценку по формуле:

$$\text{оценка} = \frac{\text{плюсы}}{\text{минусы} + \text{плюсы}} * 10 = \frac{6}{11} * 10 \approx 0.55$$

Путем округления, получим итоговую оценку: **6**