

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Уфимский Государственный Авиационный Технический Университет

Кафедра АСУ

РАСЧЕТНО-ГРАФИЧЕСКАЯ РАБОТА ПО ДИСЦИПЛИНЕ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ МОДЕЛИРОВАНИЕ И
ПРОГРАММИРОВАНИЕ»

«Депозитный калькулятор»

Выполнил:
ст. гр ПИ-209з
Марков Н.Г

Проверил:
Старцев Г.В.

Уфа, 2019 г.

Содержание

| | |
|--|----------|
| 1. Описание прототипа программного продукта | 3 |
| 2. Программный код | 4 |
| 3. Описание тестирования работы программы. | 5 |
| 4. Приложение | 6 |
| 4.1. Deposit.java - основной класс | 6 |
| 4.2. Test.java - unit-тесты | 8 |
| 4.3. Proc.java - эnumератор | 9 |
| 4.4. wind.java - GUI и ввод | 10 |

1. Описание прототипа программного продукта

Для создания депозитного калькулятора была выбрана среда выполнения Eclipse; Для работы с визуальными формами было установлено официальное расширение WindowBuilder, которое добавляет визуальную форму в которой можно редактировать Swing компоненты.

Были определены входные и выходные данные для депозитного калькулятора:

- Сумма вклада
- Дата открытия
- Процентная ставка по вкладу
- Срок вклада
- Капитализация вклада
- Периодичность капитализации

Из выходных данных требовались:

- Итоговый депозит
- Отдельно количество процентов
- Опционально таблица с периодами выплат

После анализа входных и выходных данных была собрана визуальная форма в WindowBuilder. Форма состоит из полей ввода для информации, чекбокса для выбора капитализации и ком-

Рис. 1. Итоговый вид формы

бокса для выбора частоты капитализации. Для того, что бы без выбора капитализации нельзя было сменить его частоту был использован следующий код:

```

1 //ОБРАБОТКА ФЛАЖКА
2 ch_capital.addActionListener(new ActionListener() {
3     public void actionPerformed(ActionEvent arg0) {
4         if (ch_capital.isSelected()) //без капитализации вывод недоступен (т.к. ничего не меняет)
5             {c_type_proc.setEnabled(true);}else
6             {c_type_proc.setEnabled(false);}
7     }
8 });

```

Так же форма содержит место для скроллящейся таблицы.

2. Программный код

Проект состоит из 4х файлов:

- Deposit.java - основной класс, получает входные данные
- Proc.java - содержит эnumератор для комбобокса
- Test.java - содержит юнит-тесты
- wind.java - основной класс UI и обработки данных

Deposit.java содержит методы

- public void calculate() - публичный метод, обрабатывает данные
- private void dayInPeriod() - считает время от от начала вклада до забора
- private double howDay() - возвращает кол-во дней в текущем месяце
- private double mathMoney() - подсчитывает проценты по формуле
- private void addCapital(double capMoney, int i) - добавляет проценты к общей сумме и вносит строку в таблицу
- private void simpleInterest() - подсчитывает простые проценты
- private void hardInterest() - подсчитывает сложные проценты

Wind.java содержит методы:

- private void initComp() - создание GUI
- private void create_event() - обработка событий
 - Нажатие на флажок - блокирует комбобокс
 - Нажатие на кнопку расчёта - проверяет данные из текстовых, в случае успеха передает их созданному экземпляру класса deposit;

Полный исходный код всех файлов приведен в приложении. Так же весь исходный код доступен в репозитории github: <https://github.com/fauls/DCalc>

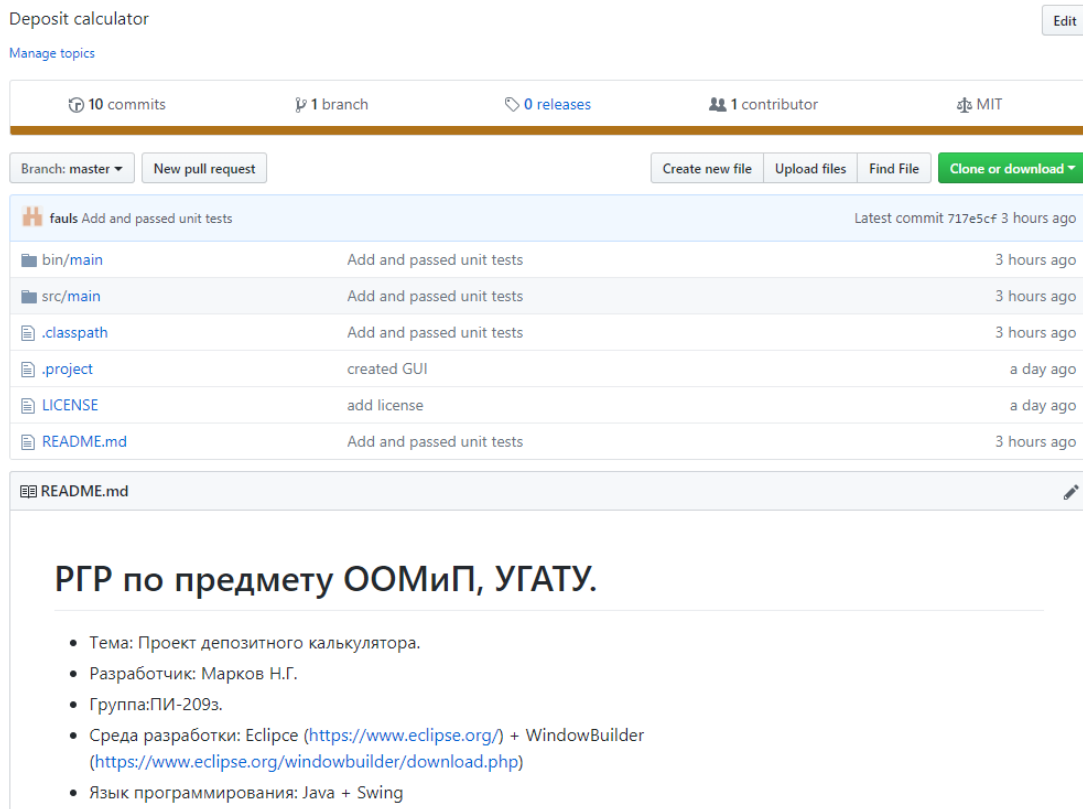


Рис. 2. Страница репозитория

3. Описание тестирования работы программы.

Для тестирования использовались JUnit5 тесты, содержащиеся в файле Test.java. В ходе тестирования создавался экземпляр класса deposit, которому передавались стартовые значения и вызывался основной метод calculate. После этого тестовой функцией assertEquals проводилось сравнение выхода программы и эталонного значения, полученного с различных онлайн-депозитных калькуляторов.

Всего было написано 6 тестовых кейсов с различными условиями, полный код юнит-тестов можно найти в приложении.

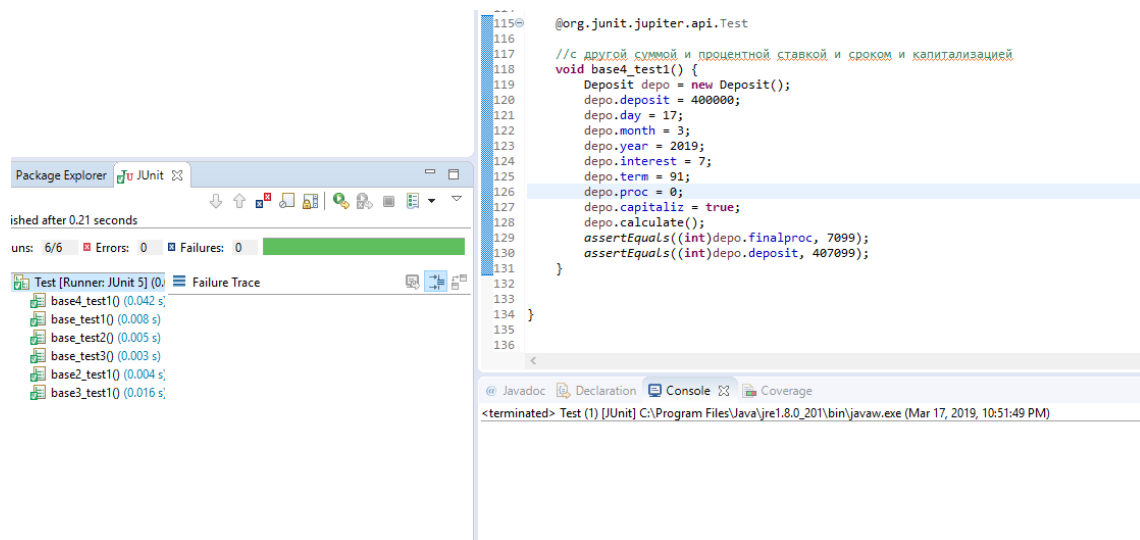


Рис. 3. Пример пройденного unit-тестирования

С помощью депозитного калькулятора вы сможете рассчитать ваш доход от размещения в банке вклада (депозита) на различные сроки и условия выплаты процентов. Рекомендуем задавать в калькуляторе условия, приближенные к рыночным на текущий момент.

Обращаясь в банк, проверьте, есть ли у него лицензия Центрального банка Российской Федерации, позволяющая привлекать вклады физических лиц, а также входит ли он в систему страхования вкладов. Помните, что застрахованная государством сумма вклада (а точнее – всех вкладов и счетов в одном банке) с учетом начисленных процентов не превышает 1,4 млн рублей.

Сумма вклада: руб. -

Процентная ставка, % годовых: ☐ Переносить даты платежей на понедельник

Дата открытия:

☒ Начисление процентов с учетом капитализации ☐ Досрочное закрытие вклада

Срок вклада: год -

Периодичность капитализации:

Дополнительные пополнения вклада

Частичное изъятие вклада

Параметры вклада

Сумма вклада: 1 000 000,00 руб.

Срок вклада: 1 год

Процентная ставка: 10,00% годовых

Дата открытия вклада: 17.03.2019

Периодичность капитализации: Ежемесячно

Порядок начисления процентов: С капитализацией

Депозитный калькулятор

Сумма вклада: Итоговая сумма:

Дата открытия:

Выплачено процентов:

Срок вклада: дней.

Процентная ставка: %

☒ Капитализация процентов

Частота капитализации:

| Дата | Проценты | Остаток |
|----------|----------|------------|
| 17/04/19 | 8493.2 | 1008493.2 |
| 17/05/19 | 8289.0 | 1016782.1 |
| 17/06/19 | 8635.7 | 1025417.8 |
| 17/07/19 | 8428.1 | 1033845.9 |
| 17/08/19 | 8780.6 | 1042626.5 |
| 17/09/19 | 8855.2 | 1051481.7 |
| 17/10/19 | 8642.3 | 1060124.0 |
| 17/11/19 | 9003.8 | 1069127.8 |
| 17/12/19 | 8780.6 | 1077908.4 |
| 17/01/20 | 8493.2 | 1086391.6 |
| 17/02/20 | 8289.0 | 1094680.6 |
| 17/03/20 | 8493.2 | 1102873.8 |
| 17/04/20 | 8289.0 | 1110962.8 |
| 17/05/20 | 8635.7 | 1119598.5 |
| 17/06/20 | 8428.1 | 1127926.6 |
| 17/07/20 | 8780.6 | 1136707.2 |
| 17/08/20 | 8855.2 | 1145562.4 |
| 17/09/20 | 8642.3 | 1154204.7 |
| 17/10/20 | 9003.8 | 1163208.5 |
| 17/11/20 | 8780.6 | 1171989.1 |
| 17/12/20 | 8493.2 | 1180482.3 |
| 17/01/21 | 8289.0 | 1188771.3 |
| 17/02/21 | 8493.2 | 1197264.5 |
| 17/03/21 | 8289.0 | 1205953.5 |
| 17/04/21 | 8493.2 | 1214846.7 |
| 17/05/21 | 8289.0 | 1223935.7 |
| 17/06/21 | 8635.7 | 1233271.4 |
| 17/07/21 | 8428.1 | 1242849.5 |
| 17/08/21 | 8780.6 | 1252670.1 |
| 17/09/21 | 8855.2 | 1262725.3 |
| 17/10/21 | 8642.3 | 1273017.6 |
| 17/11/21 | 9003.8 | 1283461.4 |
| 17/12/21 | 8780.6 | 1294142.0 |
| 17/01/22 | 8493.2 | 1305035.2 |
| 17/02/22 | 8289.0 | 1316124.2 |
| 17/03/22 | 8493.2 | 1327417.4 |
| 17/04/22 | 8289.0 | 1338806.4 |
| 17/05/22 | 8635.7 | 1350392.1 |
| 17/06/22 | 8428.1 | 1362170.2 |
| 17/07/22 | 8780.6 | 1374150.8 |
| 17/08/22 | 8855.2 | 1386336.0 |
| 17/09/22 | 8642.3 | 1398728.3 |
| 17/10/22 | 9003.8 | 1411332.1 |
| 17/11/22 | 8780.6 | 1424132.7 |
| 17/12/22 | 8493.2 | 1437125.9 |
| 17/01/23 | 8289.0 | 1450314.9 |
| 17/02/23 | 8493.2 | 1463708.1 |
| 17/03/23 | 8289.0 | 1477297.1 |
| 17/04/23 | 8493.2 | 1491190.3 |
| 17/05/23 | 8289.0 | 1505379.3 |
| 17/06/23 | 8635.7 | 1519815.0 |
| 17/07/23 | 8428.1 | 1534493.1 |
| 17/08/23 | 8780.6 | 1549413.7 |
| 17/09/23 | 8855.2 | 1564578.9 |
| 17/10/23 | 8642.3 | 1579991.2 |
| 17/11/23 | 9003.8 | 1595745.0 |
| 17/12/23 | 8780.6 | 1611825.6 |
| 17/01/24 | 8493.2 | 1628218.8 |
| 17/02/24 | 8289.0 | 1644907.8 |
| 17/03/24 | 8493.2 | 1661891.0 |
| 17/04/24 | 8289.0 | 1679170.0 |
| 17/05/24 | 8635.7 | 1696745.7 |
| 17/06/24 | 8428.1 | 1714523.8 |
| 17/07/24 | 8780.6 | 1732504.4 |
| 17/08/24 | 8855.2 | 1750689.6 |
| 17/09/24 | 8642.3 | 1769071.9 |
| 17/10/24 | 9003.8 | 1787755.7 |
| 17/11/24 | 8780.6 | 1806736.3 |
| 17/12/24 | 8493.2 | 1826019.5 |
| 17/01/25 | 8289.0 | 1845603.5 |
| 17/02/25 | 8493.2 | 1865496.7 |
| 17/03/25 | 8289.0 | 1885685.7 |
| 17/04/25 | 8493.2 | 1906178.9 |
| 17/05/25 | 8289.0 | 1926972.1 |
| 17/06/25 | 8635.7 | 1948067.8 |
| 17/07/25 | 8428.1 | 1969465.9 |
| 17/08/25 | 8780.6 | 1991166.5 |
| 17/09/25 | 8855.2 | 2013171.7 |
| 17/10/25 | 8642.3 | 2035484.0 |
| 17/11/25 | 9003.8 | 2058097.8 |
| 17/12/25 | 8780.6 | 2080918.4 |
| 17/01/26 | 8493.2 | 2104041.6 |
| 17/02/26 | 8289.0 | 2127460.6 |
| 17/03/26 | 8493.2 | 2151173.8 |
| 17/04/26 | 8289.0 | 2175182.8 |
| 17/05/26 | 8635.7 | 2199498.5 |
| 17/06/26 | 8428.1 | 2224126.6 |
| 17/07/26 | 8780.6 | 2249067.2 |
| 17/08/26 | 8855.2 | 2274322.4 |
| 17/09/26 | 8642.3 | 2299894.7 |
| 17/10/26 | 9003.8 | 2325798.5 |
| 17/11/26 | 8780.6 | 2352029.1 |
| 17/12/26 | 8493.2 | 2378582.3 |
| 17/01/27 | 8289.0 | 2405461.3 |
| 17/02/27 | 8493.2 | 2432674.5 |
| 17/03/27 | 8289.0 | 2460213.5 |
| 17/04/27 | 8493.2 | 2488086.7 |
| 17/05/27 | 8289.0 | 2516295.7 |
| 17/06/27 | 8635.7 | 2544831.4 |
| 17/07/27 | 8428.1 | 2573699.5 |
| 17/08/27 | 8780.6 | 2602900.1 |
| 17/09/27 | 8855.2 | 2632435.3 |
| 17/10/27 | 8642.3 | 2662307.6 |
| 17/11/27 | 9003.8 | 2692511.4 |
| 17/12/27 | 8780.6 | 2723047.0 |
| 17/01/28 | 8493.2 | 2753915.2 |
| 17/02/28 | 8289.0 | 2785124.2 |
| 17/03/28 | 8493.2 | 2816677.4 |
| 17/04/28 | 8289.0 | 2848576.4 |
| 17/05/28 | 8635.7 | 2880822.1 |
| 17/06/28 | 8428.1 | 2913420.2 |
| 17/07/28 | 8780.6 | 2946370.8 |
| 17/08/28 | 8855.2 | 2979676.0 |
| 17/09/28 | 8642.3 | 3013338.3 |
| 17/10/28 | 9003.8 | 3047362.1 |
| 17/11/28 | 8780.6 | 3081742.7 |
| 17/12/28 | 8493.2 | 3116475.9 |
| 17/01/29 | 8289.0 | 3151569.1 |
| 17/02/29 | 8493.2 | 3187012.3 |
| 17/03/29 | 8289.0 | 3222801.3 |
| 17/04/29 | 8493.2 | 3258934.5 |
| 17/05/29 | 8289.0 | 3295423.5 |
| 17/06/29 | 8635.7 | 3332269.2 |
| 17/07/29 | 8428.1 | 3369477.3 |
| 17/08/29 | 8780.6 | 3407057.9 |
| 17/09/29 | 8855.2 | 3445013.1 |
| 17/10/29 | 8642.3 | 3483355.4 |
| 17/11/29 | 9003.8 | 3522089.2 |
| 17/12/29 | 8780.6 | 3561214.8 |
| 17/01/30 | 8493.2 | 3600732.0 |
| 17/02/30 | 8289.0 | 3640641.0 |
| 17/03/30 | 8493.2 | 3680944.2 |
| 17/04/30 | 8289.0 | 3721643.2 |
| 17/05/30 | 8635.7 | 3762738.9 |
| 17/06/30 | 8428.1 | 3804230.0 |
| 17/07/30 | 8780.6 | 3846110.6 |
| 17/08/30 | 8855.2 | 3888480.8 |
| 17/09/30 | 8642.3 | 3931343.1 |
| 17/10/30 | 9003.8 | 3974696.9 |
| 17/11/30 | 8780.6 | 4018542.5 |
| 17/12/30 | 8493.2 | 4062879.7 |
| 17/01/31 | 8289.0 | 4107708.7 |
| 17/02/31 | 8493.2 | 4153031.9 |
| 17/03/31 | 8289.0 | 4198850.9 |
| 17/04/31 | 8493.2 | 4245165.1 |
| 17/05/31 | 8289.0 | 4291974.1 |
| 17/06/31 | 8635.7 | 4339279.8 |
| 17/07/31 | 8428.1 | 4387081.9 |
| 17/08/31 | 8780.6 | 4435382.5 |
| 17/09/31 | 8855.2 | 4484183.7 |
| 17/10/31 | 8642.3 | 4533486.0 |
| 17/11/31 | 9003.8 | 4583290.8 |
| 17/12/31 | 8780.6 | 4633597.4 |
| 17/01/32 | 8493.2 | 4684405.6 |
| 17/02/32 | 8289.0 | 4735714.6 |
| 17/03/32 | 8493.2 | 4787527.8 |
| 17/04/32 | 8289.0 | 4839846.8 |
| 17/05/32 | 8635.7 | 4892672.5 |
| 17/06/32 | 8428.1 | 4946000.6 |
| 17/07/32 | 8780.6 | 5000831.2 |
| 17/08/32 | 8855.2 | 5056166.4 |
| 17/09/32 | 8642.3 | 5112007.7 |
| 17/10/32 | 9003.8 | 5168356.5 |
| 17/11/32 | 8780.6 | 5225213.1 |
| 17/12/32 | 8493.2 | 5282576.3 |
| 17/01/33 | 8289.0 | 5340445.3 |
| 17/02/33 | 8493.2 | 5398828.5 |
| 17/03/33 | 8289.0 | 5457717.5 |
| 17/04/33 | 8493.2 | 5517110.7 |
| 17/05/33 | 8289.0 | 5577007.7 |
| 17/06/33 | 8635.7 | 5637408.4 |
| 17/07/33 | 8428.1 | 5698312.5 |
| 17/08/33 | 8780.6 | 5759720.1 |
| 17/09/33 | 8855.2 | 5821632.3 |
| 17/10/33 | 8642.3 | 5884049.6 |
| 17/11/33 | 9003.8 | 5947073.4 |
| 17/12/33 | 8780.6 | 6010704.0 |
| 17/01/34 | 8493.2 | 6074942.2 |
| 17/02/34 | 8289.0 | 6139789.2 |
| 17/03/34 | 8493.2 | 6205242.4 |
| 17/04/34 | 8289.0 | 6271301.4 |
| 17/05/34 | 8635.7 | 6337967.1 |
| 17/06/34 | 8428.1 | 6405239.2 |
| 17/07/34 | 8780.6 | 6473119.8 |
| 17/08/34 | 8855.2 | 6541610.0 |
| 17/09/34 | 8642.3 | 6610722.3 |
| 17/10/34 | 9003.8 | 6680456.1 |
| 17/11/34 | 8780.6 | 6750811.7 |
| 17/12/34 | 8493.2 | 6821798.9 |
| 17/01/35 | 8289.0 | 6893417.1 |
| 17/02/35 | 8493.2 | 6965666.3 |
| 17/03/35 | 8289.0 | 7038545.3 |
| 17/04/35 | 8493.2 | 7112054.5 |
| 17/05/35 | 8289.0 | 7186203.5 |
| 17/06/35 | 8635.7 | 7261003.2 |
| 17/07/35 | 8428.1 | 7336453.3 |
| 17/08/35 | 8780.6 | 7412553.9 |
| 17/09/35 | 8855.2 | 7489309.1 |
| 17/10/35 | 8642.3 | 7566721.4 |
| 17/11/35 | 9003.8 | 7644792.2 |
| 17/12/35 | 8780.6 | 7723523.8 |
| 17/01/36 | 8493.2 | 7802917.0 |
| 17/02/36 | 8289.0 | 7882972.0 |
| 17/03/36 | 8493.2 | 7963689.2 |
| 17/04/36 | 8289.0 | 8045068.2 |
| 17/05/36 | 8635.7 | 8127109.9 |
| 17/06/36 | 8428.1 | 8209812.0 |
| 17/07/36 | 8780.6 | 8293172.6 |
| 17/08/36 | 8855.2 | 8377192.8 |
| 17/09/36 | 8642.3 | 8461875.1 |
| 17/10/36 | 9003.8 | 8547218.9 |
| 17/11/36 | 8780.6 | 8633223.5 |
| 17/12/36 | 8493.2 | 8719896.7 |
| 17/01/37 | 8289.0 | 8807237.7 |
| 17/02/37 | 8493.2 | 8895246.9 |
| 17/03/37 | 8289.0 | 8983925.9 |
| 17/04/37 | 8493.2 | 9073274.1 |
| 17/05/37 | 8289.0 | 9163293.1 |
| 17/06/37 | 8635.7 | 9253982.8 |
| 17/07/37 | 8428.1 | 9345340.9 |
| 17/08/37 | 8780.6 | 9437367.5 |
| 17/09/37 | 8855.2 | 9530062.7 |
| 17/10/37 | 8642.3 | 9623425.0 |
| 17/11/37 | 9003.8 | 9717456.8 |
| 17/12/37 | 8780.6 | 9812158.4 |
| 17/01/38 | 8493.2 | 9907530.6 |
| 17/02/38 | 8289.0 | 10003572.6 |
| 17/03/38 | 8493.2 | 10100285.8 |
| 17/04/38 | 8289.0 | 10197669.8 |
| 17/05/38 | 8635.7 | 10295725.5 |
| 17/06/38 | 8428.1 | 10394453.6 |
| 17/07/38 | 8780.6 | 10493854.2 |
| 17/08/38 | 8855.2 | 10593929.4 |
| 17/09/38 | 8642.3 | 10694671.7 |
| 17/10/38 | 9003.8 | 10796085.5 |
| 17/11/38 | 8780.6 | 10898170.1 |
| 17/12/38 | 8493.2 | 10999923.3 |
| 17/01/39 | 8289.0 | 11102346.3 |
| 17/02/39 | 8493.2 | 11205439.5 |
| 17/03/39 | 8289.0 | 11309202.5 |
| 17/04/39 | 8493.2 | 11413635.7 |
| 17/05/39 | 8289.0 | 11518738.7 |
| 17/06/39 | 8635.7 | 11624514.4 |
| 17/07/39 | 8428.1 | 11730962.5 |
| 17/08/39 | 8780.6 | 11838083.1 |
| 17/09/39 | 8855.2 | 11945878.3 |
| 17/10/39 | 8642.3 | 12054340.6 |
| 17/11/39 | 9003.8 | 12163472.4 |
| 17/12/39 | 8780.6 | 12273273.0 |
| 17/01/40 | 8493.2 | 12383743.2 |
| 17/02/40 | 8289.0 | 12494882.2 |
| 17/03/40 | 8493.2 | 12606691.0 |
| 17/04/40 | 8289.0 | 12719169.0 |
| 17/05/40 | 8635.7 | 12832317.7 |
| 17/06/40 | 8428.1 | 12946145.8 |
| 17/07/40 | 8780.6 | 13060652.4 |
| 17/08/40 | 8855.2 | 13175837.6 |
| 17/09/40 | 8642.3 | 13291700.9 |
| 17/10/40 | 9003.8 | 13408243.7 |
| 17/11/40 | 8780.6 | 13525464.3 |
| 17/12/40 | 8493.2 | 13643362.5 |
| 17/01/41 | 8289.0 | 13761937.5 |
| 17/02/41 | 8493.2 | 13881180.7 |
| 17/03/41 | 8289.0 | 13999991.7 |
| 17/04/41 | 8493.2 | 14119360.9 |
| 17/05/41 | 8289.0 | 14239298.9 |
| 17/06/41 | 8635.7 | 14359805.6 |
| | | |

```

20
21 private LocalDate _startdate; //начальная дата
22 private LocalDate _stopdate; //конечная
23 private int _periods; //кол-во месяцев (проценты добавляются ежемесячно)
24 double finalproc=0; //итоговые проценты
25
26 public void calculate()
27 {
28     dayInPeriod();//подсчёт срока
29
30     if (capitaliz)
31     {hardInterest();}
32     else
33     {simpleInterest();}//если не капитализация, то это простые проценты
34 }
35
36 private void dayInPeriod()//считаем время от вложения вклада до забора
37 {
38     _startdate = LocalDate.of(year, month, day); //конвертируем дату из формата ввода в удобный
39     _stopdate = _startdate;
40     _stopdate = _stopdate.plusDays(term+1);//начальная дата + срок = финальная дата +1 т.к финальная не
41     учитывается
42     _periods = (int) ChronoUnit.MONTHS.between(_startdate, _stopdate);//кол-во полных месяцев в периоде
43 }
44
45 private double howDay()//сколько дней в месяце
46 {
47     return _startdate.lengthOfMonth();
48 }
49
50 private double mathMoney()//подсчёт по формуле
51 {
52     return deposit*((interest/100)*(howDay())/_startdate.lengthOfYear()); //подневный расчет
53 }
54
55 private void addCapital(double capMoney, int i)//добавить сумму к финальной, внести строку в таблицу,
56 сдвинуть дату
57 {
58     finalproc += capMoney; //финальные проценты - сумма всех процентов
59     switch(proc)//добавление месяцев, в зависимости от частоты вывода в таблицу
60     {
61         case 0: _startdate = _startdate.plusMonths(1); break;//добавление месяца
62         case 1: _startdate = _startdate.plusMonths(3); break;
63         case 2: _startdate = _startdate.plusYears(1); break;
64     }
65     table[i][0]=_startdate.format(DateTimeFormatter.ofPattern("d/MM/YY")); //форматированный вывод даты в
66     первую колонку
67     table[i][1]=String.format("%.1f",capMoney); //формат до 2х знаков после запятой
68     table[i][2]=String.format("%.1f",deposit);
69 }
70
71 private void simpleInterest()//простые проценты, для них выдача/невыдача денег не важна, так что проц
72 игнорируется
73 {
74     table = new String[_periods][3]; //инициализация таблицы
75
76     for(int i=0; i<_periods;i++) //добавление процентов каждый период
77     {addCapital(mathMoney(), i); //считаем
78     }
79     deposit+=finalproc; //к итогу добавляем проценты
80 }
81
82 private void hardInterest()//сложные проценты
83 {
84     table = new String[_periods][3]; //инициализация таблицы
85     double capMoney = 0; //накопление процентов за период
86     int clock = 0; //для отсчёта периодов
87     int period =0; //для ровного вывода строк
88     for(int i=0; i<_periods;i++) //добавление процентов каждый период
89     {
90         capMoney+=mathMoney(); //накапливать за период
91         clock++; //отсчёт периодов
92
93         switch(proc)//проведение капитализации в зависимости от выбранного варианта
94         {
95             case 0: if (clock==1) {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
96                 =0; clock=0;} break;

```

```

93         case 1: if (clock==3) {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
94             =0; clock=0;} break;
95         case 2: if (clock==12) {deposit +=capMoney; addCapital(capMoney, period); period++; capMoney
96             =0; clock=0;} break;
97     }
98 }
99
100 }

```

4.2. Test.java - unit-тесты

Листинг 3. UNIT-тесты

```

1  /**
2   *
3   */
4  package main;
5
6  //import static org.junit.jupiter.api.Assertions.*;
7  //import org.junit.Test;
8  //import org.junit.After;
9  //import org.junit.Before;
10
11 import static org.junit.jupiter.api.Assertions.assertEquals;
12
13 //import org.junit.After;
14 //import org.junit.Before;
15 /**
16  * @author stalk
17  *
18  */
19 class Test {
20
21     @org.junit.jupiter.api.Test
22
23     //первый тест без капитализации, с первым калькулятором
24     //вводе известен
25     //ожидается вывод 100 229 руб в проценты
26     //1 100 229 всего
27     //ожидаемые числа взяты как мода (самое часто встречающееся значение) из онлайн-калькуляторов
28     void base_test1() {
29         Deposit depo = new Deposit();
30         depo.deposit = 1000000;
31         depo.day = 17;
32         depo.month = 3;
33         depo.year = 2019;
34         depo.interest = 10;
35         depo.term = 365;
36         depo.proc = 0;
37         depo.capitaliz = false;
38         depo.calculate();
39         assertEquals((int)depo.finalproc, 100229); //преобразование до int из за погрешностей
40         assertEquals((int)depo.deposit, 1100229);
41     }
42
43     @org.junit.jupiter.api.Test
44
45     //аналогично, только теперь с капитализацией
46     void base_test2() {
47         Deposit depo = new Deposit();
48         depo.deposit = 1000000;
49         depo.day = 17;
50         depo.month = 3;
51         depo.year = 2019;
52         depo.interest = 10;
53         depo.term = 365;
54         depo.proc = 0;
55         depo.capitaliz = true;
56         depo.calculate();
57         assertEquals((int)depo.finalproc, 104963);
58         assertEquals((int)depo.deposit, 1104963);
59     }
60
61     @org.junit.jupiter.api.Test

```



```

62
63 //аналогично, только теперь с капитализацией по периодам
64 void base_test3() {
65     Deposit depo = new Deposit();
66     depo.deposit = 1000000;
67     depo.day = 17;
68     depo.month = 3;
69     depo.year = 2019;
70     depo.interest = 10;
71     depo.term = 365;
72     depo.proc = 1;
73     depo.capitaliz = true;
74     depo.calculate();
75     assertEquals((int)depo.finalproc, 104107);
76     assertEquals((int)depo.deposit, 1104107);
77 }
78
79 @org.junit.jupiter.api.Test
80
81 //с другой суммой и процентной ставкой
82 void base2_test1() {
83     Deposit depo = new Deposit();
84     depo.deposit = 2000000;
85     depo.day = 17;
86     depo.month = 3;
87     depo.year = 2019;
88     depo.interest = 8;
89     depo.term = 365;
90     depo.proc = 0;
91     depo.capitaliz = false;
92     depo.calculate();
93     assertEquals((int)depo.finalproc, 160366);
94     assertEquals((int)depo.deposit, 2160366);
95 }
96
97 @org.junit.jupiter.api.Test
98
99 //с другой суммой и процентной ставкой и сроком
100 void base3_test1() {
101     Deposit depo = new Deposit();
102     depo.deposit = 500000;
103     depo.day = 17;
104     depo.month = 3;
105     depo.year = 2019;
106     depo.interest = 5;
107     depo.term = 60;
108     depo.proc = 0;
109     depo.capitaliz = false;
110     depo.calculate();
111     assertEquals((int)depo.finalproc, 4178);
112     assertEquals((int)depo.deposit, 504178);
113 }
114
115 @org.junit.jupiter.api.Test
116
117 //с другой суммой и процентной ставкой и сроком и капитализацией
118 void base4_test1() {
119     Deposit depo = new Deposit();
120     depo.deposit = 400000;
121     depo.day = 17;
122     depo.month = 3;
123     depo.year = 2019;
124     depo.interest = 7;
125     depo.term = 91;
126     depo.proc = 0;
127     depo.capitaliz = true;
128     depo.calculate();
129     assertEquals((int)depo.finalproc, 7099);
130     assertEquals((int)depo.deposit, 407099);
131 }
132
133
134 }

```

4.3. Proc.java - эnumератор

```

1 package main;
2 //эnumератор для комбобокса
3 public enum Proc {
4     ежемесячно,
5     ежеквартально,
6     ежегодно
7 }

```

4.4. wind.java - GUI и ввод

```

1 package main;
2
3 import java.awt.EventQueue;
4
5 import javax.swing.JFrame;
6 import javax.swing.JPanel;
7 import javax.swing.border.EmptyBorder;
8 import javax.swing.GroupLayout;
9 import javax.swing.GroupLayout.Alignment;
10 import javax.swing.JLabel;
11 import javax.swing.JOptionPane;
12 import javax.swing.JTextField;
13 import javax.swing.LayoutStyle.ComponentPlacement;
14 import javax.swing.UIManager;
15 import javax.swing.JComboBox;
16 import javax.swing.DefaultComboBoxModel;
17
18 import java.time.LocalDate;
19 import java.time.Month;
20 import javax.swing.JCheckBox;
21 import javax.swing.JButton;
22 import javax.swing.JTextPane;
23 import java.awt.event.ActionListener;
24 import java.awt.event.ActionEvent;
25 import javax.swing.JTable;
26 import javax.swing.table.DefaultTableModel;
27 import javax.swing.JScrollPane;
28
29 public class wind extends JFrame {
30
31     private JPanel contentPane;
32     private JTextField t_sum;
33     private JTextField t_day;
34     private JTextField t_year;
35     private JTextField t_month;
36     private JLabel l_time_vklad;
37     private JTextField t_time_vklad;
38     private JLabel label;
39     private JLabel l_proc;
40     private JTextField t_proc;
41     private JLabel label_2;
42     private JLabel label_3;
43     private JLabel label_4;
44     private JButton b_go;
45     private JCheckBox ch_capital;
46     private JTable table;
47     private JTextPane t_prok_out;
48     private JTextPane t_ost_vklad;
49     private JComboBox c_type_proc;
50
51     /**
52      * Launch the application.
53      */
54     public static void main(String[] args) {
55         try {
56             UIManager.setLookAndFeel("javax.swing.plaf.nimbus.NimbusLookAndFeel");
57         } catch (Throwable e) {
58             e.printStackTrace();
59         }
60         EventQueue.invokeLater(new Runnable() {
61             public void run() {
62                 try {
63                     wind frame = new wind();

```

```

64         frame.setVisible(true);
65     } catch (Exception e) {
66         e.printStackTrace();
67     }
68     }
69     });
70 }
71
72 /**
73  * Create the frame.
74  */
75 public wind() { //главная функция
76     initComp(); //создание компонентов
77     create_event(); //обработка событий
78 }
79
80 //
81 //ОБРАБОТКА СОБЫТИЙ
82 //
83 private void create_event()
84 {
85
86     //ОБРАБОТКА ФЛАЖКА
87     ch_capital.addActionListener(new ActionListener() {
88         public void actionPerformed(ActionEvent arg0) {
89             if (ch_capital.isSelected()) //без капитализации вывод недоступен (т.к. ничего не меняет
90                 )
91                 {c_type_proc.setEnabled(true);}else
92                 {c_type_proc.setEnabled(false);}
93         }
94     });
95
96     //ОБРАБОТКА КНОПКИ
97     b_go.addActionListener(new ActionListener() {
98         public void actionPerformed(ActionEvent arg0) { //нажатие на кнопку "Рассчитать"
99             Deposit deposit = new Deposit(); //создаём новый класс депозита
100             //ввод значений и проверка
101
102             //
103             //ОБРАБОТКА ВВОДА
104             //
105
106             try{//сумма вклада, переменная deposit
107                 deposit.deposit = Integer.parseInt(t_sum.getText());
108             } catch (NumberFormatException e) {
109                 JOptionPane.showMessageDialog(null, "Неверно_указана_сумма_вклада");
110             }
111
112             try{//день
113                 int day = Integer.parseInt(t_day.getText());
114
115                 if (day<1 || day>31) {throw new NumberFormatException();}else//генерируем
116                     исключение если выходим за пределы
117                     {deposit.day = day;}
118             } catch (NumberFormatException e) {
119                 JOptionPane.showMessageDialog(null, "Введите_день_числом_от_1_до_31");
120             }
121             try{//месяц
122                 int month = Integer.parseInt(t_month.getText());
123
124                 if (month<1 || month>12) {throw new NumberFormatException();}else
125                     {deposit.month = month;}
126             } catch (NumberFormatException e) {
127                 JOptionPane.showMessageDialog(null, "Введите_месяц_числом_от_1_до_12");
128             }
129             try{//год
130                 int year = Integer.parseInt(t_year.getText());
131
132                 if (year<0) {throw new NumberFormatException();}else
133                     {deposit.year = year;}
134             } catch (NumberFormatException e) {
135                 JOptionPane.showMessageDialog(null, "Введите_год_числом_больше_0");
136             }
137
138             try{//срок вклада
139                 int term = Integer.parseInt(t_time_vklad.getText());

```

```

141
142         if (term<0) {throw new NumberFormatException();}else
143         {deposit.term = term;}
144
145         } catch (NumberFormatException e) {
146             JOptionPane.showMessageDialog(null, "Введите_срок_числом_больше_0");
147         }
148         try{//процентная ставка
149             int interest = Integer.parseInt(t_proc.getText());
150
151             if (interest<0) {throw new NumberFormatException();}else
152             {deposit.interest = interest;}
153
154             } catch (NumberFormatException e) {
155                 JOptionPane.showMessageDialog(null, "Введите_число_без_знака_процента");
156             }
157
158             deposit.capitaliz = ch_capital.isSelected(); //капитализация
159             deposit.proc = c_type_proc.getSelectedIndex();//капитализация, 0 - ежемесячно, 1 -
                ежеквартально, 2 - ежегодно
160
161             //
162             //ВЫЧИСЛЕНИЯ
163             //
164             deposit.calculate();//вычисления
165
166             //
167             //ВЫВОД ИНФОРМАЦИИ
168             //
169             //текстбоксы
170             t_prok_out.setText(String.format("%.2f",deposit.finalproc));//форматный вывод
171             t_ost_vklad.setText(String.format("%.2f", deposit.deposit));
172             //таблица
173             table.setModel(new DefaultTableModel(//инициализация таблицы
174                 deposit.table,//двухмерным массивом из deposit
175                 new String[] {
176                     "Дата", "Проценты", "Остаток" //и заголовком
177                 }
178             ));
179             deposit = null;//очистка класса, на всякий случай
180         }
181
182     });
183 }
184
185
186 //
187 //ИНИЦИАЛИЗАЦИЯ
188 //(по большей части код из конструктора)
189
190 private void initComp()
191 {
192     setTitle("\u0414\u0435\u043c\u043e\u0432\u0435\u0434\u0435\u043d\u0438\u0442\u0435\u043c\u043e\u0439_\u0430\u0432\u0430\u043d\u0442\u0430\u0432\u0430\u043d\u0438\u044f");
193     setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
194     setBounds(100, 100, 599, 300);
195     contentPane = new JPanel();
196     contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
197     setContentPane(contentPane);
198
199     JLabel l_sum = new JLabel("\u0421\u0443\u043c\u043c\u0430_\u0430\u0432\u0430\u043d\u0438\u0442\u0430\u0432\u0430\u043d\u0438\u044f:");
200
201     t_sum = new JTextField();
202     t_sum.setToolTipText("\u0414\u0435\u043c\u043e\u0432\u0435\u0434\u0435\u043d\u0438\u0442\u0435\u043c\u043e\u0439_\u0430\u0432\u0430\u043d\u0442\u0430\u0432\u0430\u043d\u0438\u044f");
203     t_sum.setColumns(10);
204
205     JLabel l_date = new JLabel("\u0414\u0430\u0442\u0430_\u0430\u0432\u0430\u0440\u0430\u0442\u0430\u0442\u0435\u043b\u044c\u043d\u0430\u044f");
206
207     //представление даты переменными
208     LocalDate date = LocalDate.now();
209
210     int year = date.getYear();
211     Month month = date.getMonth();
212     int day = date.getDayOfMonth();
213
214     //вывод текущей даты в текстбоксы
215     t_day = new JTextField();
216     t_day.setText(""+day);

```

[illegible]

```

285         .addGroup(gl_contentPane.createParallelGroup(Alignment.
286             LEADING)
287             .addGroup(gl_contentPane.createSequentialGroup()
288                 .addComponent(l_sum)
289                 .addPreferredGap(ComponentPlacement.RELATED)
290                 .addComponent(t_sum, GroupLayout.
291                     DEFAULT_SIZE, 151, Short.MAX_VALUE))
292             .addGroup(gl_contentPane.createSequentialGroup()
293                 .addComponent(l_date)
294                 .addPreferredGap(ComponentPlacement.RELATED)
295                 .addGroup(gl_contentPane.createParallelGroup
296                     (Alignment.LEADING)
297                     .addGroup(gl_contentPane.
298                         createSequentialGroup()
299                         .addComponent(t_day,
300                             GroupLayout.
301                                 PREFERRED_SIZE, 30,
302                                 GroupLayout.
303                                     PREFERRED_SIZE)
304                         .addPreferredGap(
305                             ComponentPlacement.
306                                 RELATED, GroupLayout.
307                                     DEFAULT_SIZE, Short.
308                                         MAX_VALUE)
309                         .addComponent(t_month,
310                             GroupLayout.
311                                 PREFERRED_SIZE, 34,
312                                 GroupLayout.
313                                     PREFERRED_SIZE))
314                         .addComponent(t_time_vklad, 0, 0,
315                             Short.MAX_VALUE))
316                     .addPreferredGap(ComponentPlacement.RELATED)
317                     .addGroup(gl_contentPane.createParallelGroup
318                         (Alignment.LEADING)
319                         .addComponent(label)
320                         .addComponent(t_year, GroupLayout.
321                             DEFAULT_SIZE, 74, Short.
322                                 MAX_VALUE))))))
323         .addGroup(gl_contentPane.createSequentialGroup()
324             .addContainerGap()
325             .addComponent(l_proc)
326             .addPreferredGap(ComponentPlacement.RELATED)
327             .addComponent(t_proc, GroupLayout.PREFERRED_SIZE, 29,
328                 GroupLayout.PREFERRED_SIZE)
329             .addPreferredGap(ComponentPlacement.RELATED)
330             .addComponent(label_2))
331         .addGroup(gl_contentPane.createSequentialGroup()
332             .addContainerGap()
333             .addComponent(ch_capital))
334         .addGroup(gl_contentPane.createSequentialGroup()
335             .addContainerGap()
336             .addComponent(label_1)
337             .addPreferredGap(ComponentPlacement.RELATED)
338             .addComponent(c_type_proc, GroupLayout.PREFERRED_SIZE,
339                 GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
340             ))
341         .addGroup(gl_contentPane.createSequentialGroup()
342             .addContainerGap()
343             .addComponent(b_go, GroupLayout.DEFAULT_SIZE, 266, Short.
344                 MAX_VALUE)
345             .addGap(18)))
346     .addGap(18)
347     .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
348         .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 277, Short.MAX_VALUE)
349         .addGroup(gl_contentPane.createSequentialGroup()
350             .addComponent(label_3)
351             .addPreferredGap(ComponentPlacement.RELATED)
352             .addComponent(t_ost_vklad, GroupLayout.DEFAULT_SIZE, 180, Short.
353                 MAX_VALUE))
354         .addGroup(gl_contentPane.createSequentialGroup()
355             .addComponent(label_4)
356             .addPreferredGap(ComponentPlacement.RELATED)
357             .addComponent(t_prok_out, GroupLayout.DEFAULT_SIZE, 140, Short.
358                 MAX_VALUE)))
359     .addContainerGap())
360 );
361 gl_contentPane.setVerticalGroup(
362     gl_contentPane.createParallelGroup(Alignment.LEADING)
363     .addGroup(gl_contentPane.createSequentialGroup()

```

```

338         .addContainerGap()
339         .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
340             .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
341                 .addComponent(l_sum)
342                 .addComponent(t_sum, GroupLayout.PREFERRED_SIZE, GroupLayout.
343                     DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
344                 .addComponent(label_3))
345             .addGroup(gl_contentPane.createParallelGroup(Alignment.TRAILING)
346                 .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
347                     .addComponent(l_date)
348                     .addComponent(t_day, GroupLayout.PREFERRED_SIZE, GroupLayout.
349                         DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
350                     .addComponent(t_year, GroupLayout.PREFERRED_SIZE, GroupLayout.
351                         DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
352                     .addComponent(t_month, GroupLayout.PREFERRED_SIZE, GroupLayout.
353                         DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
354                     .addComponent(label_4))
355                 .addComponent(t_prok_out, GroupLayout.PREFERRED_SIZE, GroupLayout.
356                     DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
357             .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
358                 .addGroup(gl_contentPane.createSequentialGroup()
359                     .addGroup(gl_contentPane.createParallelGroup(
360                         Alignment.BASELINE)
361                         .addComponent(l_time_vklad)
362                         .addComponent(t_time_vklad, GroupLayout.
363                             PREFERRED_SIZE, GroupLayout.
364                             DEFAULT_SIZE, GroupLayout.
365                             PREFERRED_SIZE))
366                     .addPreferredGap(ComponentPlacement.RELATED)
367                     .addGroup(gl_contentPane.createParallelGroup(
368                         Alignment.BASELINE)
369                         .addComponent(l_proc)
370                         .addComponent(t_proc, GroupLayout.
371                             PREFERRED_SIZE, GroupLayout.
372                             DEFAULT_SIZE, GroupLayout.
373                             PREFERRED_SIZE)
374                         .addComponent(label_2)))
375                     .addComponent(label))
376                 .addPreferredGap(ComponentPlacement.RELATED)
377                 .addComponent(ch_capital)
378                 .addPreferredGap(ComponentPlacement.RELATED)
379                 .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
380                     .addComponent(label_1)
381                     .addComponent(c_type_proc, GroupLayout.PREFERRED_SIZE,
382                         GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE)
383                     )
384                 .addPreferredGap(ComponentPlacement.RELATED)
385                 .addComponent(b_go))
386             .addGroup(gl_contentPane.createSequentialGroup()
387                 .addPreferredGap(ComponentPlacement.RELATED)
388                 .addComponent(scrollPane, GroupLayout.DEFAULT_SIZE, 171, Short.
389                     MAX_VALUE)
390                 .addContainerGap()))))
391     );
392
393     table = new JTable();
394     scrollPane.setViewportViewView(table);
395     table.setShowVerticalLines(true);
396     table.setRowSelectionAllowed(false);
397     contentPane.setLayout(gl_contentPane);
398 }

```